

An Investigation of New Methods for Estimating Parameter Sensitivities

Todd J. Beltracchi
Gary A. Gabriele

Department of Mechanical Engineering,
Aeronautical Engineering & Mechanics
Rensselaer Polytechnic Institute
Troy, N.Y. 12180

August 1988

(NASA-CR-183195) AN INVESTIGATION OF NEW
METHODS FOR ESTIMATING PARAMETER
SENSITIVITIES (Rensselaer Polytechnic
Inst.) 138 p CSCL 12A

N89-10586

G3/64 Unclass
0161101

Abstract	iv
1. Introduction	1
1.1. Standard notation	1
2. Background	3
2.1. Review of Parameter Sensitivity Methods	3
2.1.1. Brute Force Methods	4
2.1.2. Kuhn-Tucker Methods	4
2.1.3. Methods Based on the Extended Design Space	6
2.1.4. Variable Sensitivities	7
2.1.5. Other Methods	8
2.2. Previous Work in Estimating Parameter Sensitivities for Changes in the Active Set	10
2.2.1. Cases to Consider	11
2.2.2. Prediction of when the Active set will Change	11
2.2.3. An Example of Case 1 and 2	12
2.2.4. Example of Case 3	16
2.3. Summary	18
3. New method for Estimating Parameter Sensitivity	20
3.1 RQP Methods	20
3.2 Proposed Algorithm for Parameter Sensitivity	23
3.3 Comparison to Existing Methods	26
3.3.1. Demonstration of Equivalence of New Method to Kuhn-Tucker Method	35
3.3.2. Performance Comparison with Other Methods	32
3.3. Potential Problems	34
3.4 Summary	35
4. Implementation	37
4.1. Modifications to RQOPT	37
4.1.1. Implementation of a Factorized BFS Variable Metric Update	37
4.1.2. Condition Number Reset	39
4.1.3. Calculation of the Lagrange Multiplier Estimates	40
4.1.4. SR1 update	41
4.2 The Creation of a System to Automatically Calculate Parameter sensitivities	42
4.2.1 The RQCRE Support System	42
4.2.2 The RQSEN program	43
5. Numerical Experiments	45
5.1. Initial Test Set	45
5.2. Convergence of the Hessian approximation	46
5.3 Results	50
5.3.1 Problem 1	51
5.3.2 Problem 2	54
5.3.3 Problem 3	56
5.3.4 Problem 4	59
5.3.1 A Graphical Comparison of the Accuracy Delivered by Several Different Methods	63
5.4. Conclusions from test results	66
6. Changes in the Active Set	68
6.1. Cases of active set change	68
6.2. Demonstration of Cases plots	69
6.3. Proposed solutions	75
6.3.1 Dealing with Case 1 and 2	76

6.3.2	An Example.....	81
6.3.3	Dealing with Case 3	82
6.4.	Generation of test Problems.....	83
6.4.1	Test Problems with Case 1 and Case 2 Changes in the Active Set.....	83
6.4.2	Generation of Test Problems for Case 3	84
7.	Conclusions and Recommendations for Future Research	86
7.1.	Conclusions	86
7.2.	Recommendations for Future Research	87
Appendix 1	Test Problems.....	91
Appendix 2	Computer Programs.....	109
7.1.	The RQCRE support System.....	109
7.2.	The RQSEN system.....	112
REFERENCES	126

1. Introduction

Estimation of the sensitivity of problem functions with respect to problem variables forms the basis for many of our modern day algorithms for engineering optimization. The most common application of problem sensitivities has been in the calculation of objective function and constraint partial derivatives for determining search directions and optimality conditions. A second form of sensitivity analysis, parameter sensitivity, has also become an important topic in recent years with the advent of renewed research in the optimization of large engineering systems by means of decomposition methods. By parameter sensitivity, we refer to the estimation of changes in the modeling functions and current design variables due to small changes in the fixed parameters of the formulation. Methods for calculating these derivatives have been proposed and have been used as the basis of a method for multi-level decomposition of large engineering problems [Sobieski, 1982]. Two drawbacks to estimating parameter sensitivities by current methods have been: (1) the need for second order information about the Lagrangian at the current point, and (2) the estimates assume no change in the active set of constraints. The objectives of this work were to investigate solutions to these two problems.

1.1. STANDARD NOTATION

To provide a framework about which we can discuss the various ways sensitivity analysis can be performed, the following standard form of the nonlinear programming problem, which explicitly represents the problem parameters, is presented.

$$\text{Minimize: } f(\mathbf{x}, \mathbf{P}) \quad \text{Objective function} \quad (1.1)$$

$$\text{Subject to: } h_l(\mathbf{x}, \mathbf{P}) = 0 \quad \text{Equality constraints } l = 1, L \quad (1.2)$$

$$g_j(\mathbf{x}, \mathbf{P}) \geq 0 \quad \text{Inequality constraints } j = 1, J \quad (1.3)$$

$$\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \quad \text{Variable bounds} \quad (1.4)$$

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \quad \text{Design variables} \quad (1.5)$$

$$\mathbf{P} = (p_1, p_2, \dots, p_k) \quad \text{Problem parameters} \quad (1.6)$$

In the above formulation, we assume that the problem functions f , g , and h can be either linear or nonlinear functions of the design variables. We also assume that the problem parameters \mathbf{P} , are held fixed during the course of the optimization. Any candidate solution point, \mathbf{x}^* , must satisfy the following first order Kuhn-Tucker conditions:

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{v}, \mathbf{u}) = 0 \quad (1.7)$$

$$h_l(\mathbf{x}) = 0 \quad l = 1, L \quad (1.8)$$

$$g_j(\mathbf{x}) \geq 0 \quad j = 1, J \quad (1.9)$$

$$u_j g_j(\mathbf{x}) = 0 \quad j = 1, J \quad (1.10)$$

$$u_j \geq 0 \quad j = 1, J \quad (1.11)$$

where the Lagrangian L , is given by:

$$L(\mathbf{x}, \mathbf{v}, \mathbf{u}) = f(\mathbf{x}) + \sum v_l h_l(\mathbf{x}) - \sum u_j g_j(\mathbf{x}) \quad (1.12)$$

At some point, usually the optimal point, we are interested in understanding the effect that changes in \mathbf{P} will have on our proposed solution \mathbf{x}^* . Therefore we seek the sensitivities, $df/d\mathbf{P}$, $\partial \mathbf{x} / \partial \mathbf{P}$, and $\partial(h, g) / \partial \mathbf{P}^1$. In this report, we will propose a new algorithm based on the Recursive Quadratic Programming (RQP) method for estimating these parameter sensitivities. The following sections provide a description of this algorithm and how it relates to current methods, a discussion of the implementation issues, and some initial testing on a test set of known characteristics. In addition, section 6 proposes some solutions for estimating sensitivities in those cases where the active set of the constraints changes when the parameter is changed.

¹ The notation (h, g) refers to the set of constraints active at the current point.

2. Background

The standard problem of parameter sensitivity analysis is to indicate how the objective function, constraints, and optimum design variables will change when problem parameters or design variables are changed from their current values. Parameter Sensitivity analysis is usually performed at a candidate optimum point where we might be interested in studying how the optimal design might be effected by changes in specifications, variability due to manufacturing, or operational noises. In this chapter we present a historical overview of the significant developments in sensitivity analysis and provide a review and assessment of current parameter sensitivity methods. The final section of the chapter reviews work done in estimating parameter sensitivities for those cases where the active constraint set changes.

2.1. REVIEW OF PARAMETER SENSITIVITY METHODS

The roots of sensitivity analysis can be traced to Lagrange (1881) when he suggested solving equality constrained extrema problems by finding the solution \mathbf{x}^* , and \mathbf{v}^* , for the equations

$$\nabla_{\mathbf{x}}L(\mathbf{x},\mathbf{v}) = 0 \quad (2.1)$$

$$h(\mathbf{x}) = 0 \quad (2.2)$$

where

$$L(\mathbf{x},\mathbf{v}) = f(\mathbf{x}) + \sum v_i h_i(\mathbf{x}) \quad (2.3)$$

where the v_i are undetermined multipliers or Lagrange multipliers. The paper did not provide the conditions for when solutions of equation (2.1-2.3), were actual solutions of the extrema problems or how to interpret the Lagrange multipliers.

Samuelson (1947) gave several interpretations of Lagrange multipliers in an economic setting. He developed approaches based on using Lagrange multipliers to solve different economic models and was the first to clearly identify Lagrange multipliers as shadow prices in an economic context. Kuhn and Tucker (1951) presented conditions for relative extrema which use the Lagrange multipliers to establish optimality (ref. eq. 1.7 - 1.12). Since 1951 several constraint qualifications and extensions to these conditions have been proposed and are described in Bazaraa and Shetty (1979).

Dantzig (1963) brought forth the idea of "Post Optimality Analysis" for linear programs. Dantzig described post optimality analysis as the calculation of the sensitivity of the optimum with respect to changes in the problem parameters. Sensitivity analysis has been widely used in linear programming, a good survey of its use is provided by Gal

(1984).

Fiacco et al. (1968,1974,1976,1983) has also done extensive research in the area of sensitivity analysis. His book "Introduction to Sensitivity and Stability Analysis" (1983) covers the significant developments in the field of sensitivity analysis prior to 1982. He has published many articles on sensitivity analysis, and has probably been the most active researcher of sensitivity analysis for nonlinear programming problems.

In the following subsections, we will discuss past work related to the determination of sensitivity information for nonlinear programming problems. The methods we will discuss range from the most simplistic approach of reoptimization to more elaborate approaches based on the Kuhn-Tucker conditions or advanced optimization methods.

2.1.1. Brute Force Methods

The simplest, and probably most used method, for parameter sensitivity analysis is to re-optimize the problem for the new values of the problem parameters and plot the trends. We will refer to this as the Brute Force method. The Brute Force method is probably the most accurate of the methods available (for large variations in Δp , but can experience round off and truncation errors when used to approximate derivatives) but it can be computationally expensive even for small problems. Examples of its use in the literature are given in Arbuckle and Sliwa (1984) and Robertson and Gabriele (1987).

Armacost and Fiacco (1974) and McKeown (1980) describe a direct approach to calculating parameter sensitivities based on the central difference approximation given below

$$\frac{df^*}{dp} = \frac{f(x^*, p + \Delta p) - f(x^*, p - \Delta p)}{2\Delta p} \quad (2.4)$$

$$\frac{\partial x^*}{\partial p} = \frac{x^*(p + \Delta p) - x^*(p - \Delta p)}{2\Delta p} \quad (2.5)$$

This method requires the problem to be reoptimized (to a high degree of accuracy) for two different values of the parameter. McKeown states that this method should not be used as a primary method for the calculation of sensitivities because it is computationally expensive.

2.1.2. Kuhn-Tucker Methods

To avoid the computational expense of reoptimization, several researchers have developed sensitivity methods based on the Kuhn-Tucker conditions (1.7) - (1.12). Two types of algorithms have resulted, those that differentiate the Kuhn-Tucker conditions with respect to p , and those that differentiate the optimality conditions for penalty functions.

In the former category, a set of Kuhn-Tucker sensitivity equations have been derived independently by several authors (Armacost and Fiacco 1974, Sobieski et. al. 1981, McKeown 1980 b) and result in the following linear system of equations.

$$\begin{bmatrix} \nabla_x^2 L & -\nabla_x(h,g) \\ \nabla_x(h,g)^T & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial p_i} \\ \frac{\partial(v,u)}{\partial p_i} \end{bmatrix} + \begin{bmatrix} \frac{\partial \nabla_x L}{\partial p_i} \\ \frac{\partial(h,g)}{\partial p_i} \end{bmatrix} = 0 \quad (2.6)$$

This linear system can be solved for the sensitivity of the design variables with respect to a problem parameter $\partial x / \partial p_i$, and the sensitivity of the Lagrange multipliers with respect to p_i , $\partial(v,u) / \partial p_i$. These can then be used to determine the sensitivity of the objective function with respect to p_i by the following

$$\frac{df}{dp_i} = \frac{\partial f}{\partial p_i} + \frac{\partial f}{\partial x}^T \frac{\partial x}{\partial p_i} \quad (2.7)$$

For any change in the parameter Δp_i , the new optimum value of the objective function or design variables can be estimated from the linear extrapolations

$$f_{\text{new}} = f(x^*_{\text{old}}) + \Delta p_i \frac{df}{dp_i} \quad (2.8)$$

$$x^*_{\text{new}} = x^*_{\text{old}} + \Delta p_i \frac{\partial x}{\partial p_i} \quad (2.9)$$

These equations are bounded by the assumption that the active set remains the same. An estimate of when the active set will change can be made by examining the Lagrange multipliers of the active inequality constraints and linear approximations of the inactive constraints. An inequality constraint should leave the active set when its Lagrange multiplier goes to zero. The corresponding value of Δp_i where this occurs is predicted by using the linear prediction

$$\Delta p_i = \frac{u_j}{\left(\frac{\partial u_j}{\partial p_i} \right)} \quad j \in \text{active set of constraints} \quad (2.10)$$

A new inequality constraint will enter the active set when its value goes to zero. A linear prediction for when this happens is given by

$$\Delta p_i = \frac{g_i}{\left(\frac{\partial g_i}{\partial p_i} + \frac{\partial g_i}{\partial x}^T \frac{\partial x}{\partial p_i} \right)} \quad j \notin \text{active set of constraints} \quad (2.11)$$

We can predict the change in active set to occur at the smallest value of Δp_i obtained from applying equations 2.10 and 2.11 to all constraints.

Fiacco (1974,1980,1983) has developed first and second order extrapolation techniques to predict the new value of the optimum when parameters are perturbed. Armacost and Fiacco have developed a second order extrapolation for the objective function value for the special case where the problem parameters are confined to being the right hand side values of the constraints. This provides second order response information for the objective function using the Lagrange multipliers and the partials with respect to P of the Lagrange multipliers.

Sobieski, et. al. (1981) observed that a more accurate estimate of f_{new} given in (2.8) can be obtained if the value of x_{new} given in (2.9) is used to calculate the value of the objective function at a perturbation Δp_i . This will be a more accurate estimate for problems where the constraints are well behaved and not highly nonlinear, but the objective function is nonlinear.

Barthelemy and Sobieski (1983) derived the following formula that can also be used to calculate the sensitivity of the objective function without the need to calculate $\partial x^*/\partial p$,

$$\frac{df^*}{dp} = \frac{\partial f}{\partial p} + \sum_{j=1}^{n_{\text{ineq}}} u_j \frac{\partial g_j}{\partial p} \quad (2.12)$$

The formula can be derived by assuming that objective function behaves like the Lagrangian in the region of the optimum. This formula has also been derived by Fiacco (1983) and McKeown (1980 b).

Diewart (1984) has developed some new sensitivity theories for dealing with the addition of constraints at the solution of economic models before the solution of the sensitivity equations. This analysis is important because there may be short term restrictions on modifications that can be made to the system. The paper presents a recursive relationship that can be used to avoid refactorizing the sensitivity equations when a new constraint is added to the problem. The paper also presents equations that can be used to calculate a second order estimate of the location of the optimum, but this formula requires third order derivatives which are seldom available in engineering.

2.1.3. Methods Based on the Extended Design Space

Vanderplaats (1984 a, 1984 b) and Vanderplaats and Yoshida (1985, 1986) have

developed an approach for calculating the sensitivity based on the method of feasible directions. The sensitivities are estimated by extending the set of design variables to include the problem parameters for which a feasible direction is then determined. This method is known as the Extended Design Space (EDS) method. Of the methods discussed, it has the dual advantages of simplicity and efficiency. Vanderplaats (1984 a) reports that the EDS method can handle near active constraints, and is able to leave constraint linearizations. However, the method does suffer from a sensitivity to one of its algorithm parameters as reported in Vanderplaats and Cai (1987), and is unable to predict when constraints will leave the active set. The EDS method is also sensitive to the restriction of the move vector to be of length one.

The EDS method can be used to assess the effect of perturbing several parameters at the same time. It is also able to solve for sensitivities of degenerate optimal points where either strict complementarity does not hold, or the constraint gradients of the active constraints are linearly dependent. The method seems to give good estimations for medium sized perturbations of the parameters, but for small perturbations the the Kuhn-Tucker method described above gives better results. Vanderplaats and Cai (1987) also report that there are some cases where the EDS algorithm can produce incorrect values of the sensitivity derivatives.

Vanderplaats also proposes a second order approximation technique which is interesting but requires second derivatives of the objective function and constraints. The second order method solves a quadratic approximating problem for a specified value of the parameter. The second order method will give good results in a larger region about the optimum than first order methods and does not appear to be as sensitive to changes in the active set as other methods are. However, there is still the problem of obtaining the Hessians of the objective function and constraints and solving the quadratic approximating problem. Vanderplaats and Cai (1987) feel that the second order EDS algorithm is the best option short of reoptimizing the problem for estimating sensitivities. But they caution that the method should not always be used because of its high computational cost.

2.1.4. Variable Sensitivities

McKeown (1980 a,c) has developed sensitivity analysis techniques for determining the sensitivity of design variables subject to perturbations about the optimum. This technique is based on an eigenvector analysis of the reduced Hessian matrix which applies to a variant of our standard problem (1.1)-(1.6) where no problem parameters exist. For unconstrained problems the major eigenvector will point in the direction of maximum increase of the objective function, the minor eigenvector will point in the direction of minimum increase of the objective function. For constrained problems the directions are

projected on the active constraints. This type of information may be useful for setting tolerances on design variables.

For McKeown's algorithm, the Hessian of the Lagrangian is needed but the analysis is performed using only the reduced Hessian of the Lagrangian. An algorithm is provided for reducing the Hessian. If the Hessian is to be evaluated numerically, an algorithm is provided for the calculation of the reduced Hessian of the Lagrangian directly. This will reduce the number of extra function evaluations that are needed to conduct the sensitivity analysis.

2.1.5. Other Work

Garcia and Zangwill (1981) describe a Homotopy approach that can be used to solve nonlinear programming problems. They state that this approach can also be used to solve parametric nonlinear programming problems and is closely related to sensitivity and perturbation analysis. Komija and Hirabay (1984) discuss some theoretical topics involved in using a Homotopy approach to calculate parameter sensitivities when the active set of constraints changes.

Dinkel and Kochenberger and Wong (1983) have developed an incremental approach for solving for the sensitivities of geometric programming problems. The approach is to ask the user for the new value of the parameter and then make several steps with corrections to reach that point. They found the smaller the step they used the more accurate the solution would be.

Jittorntrum (1984) examines solving for the sensitivity of degenerate optimum points using the Kuhn-Tucker sensitivity equations. He provides a way to solve these problems using directional derivatives which provides different answers for both positive and negative perturbations in the parameters. Other theoretical issues for the use of directional derivatives to calculate optimum parameter sensitivities have been addressed by Janin (1984), Gauvin and Dubeau (1983), and Rockafellar, R. T. (1984).

Zolezzi (1985) examines the conditions under which the Lagrange multipliers are continuous under perturbations in the problem data. This is important because Kuhn-Tucker sensitivity analysis uses Lagrange multipliers and rates of change of the Lagrange multipliers to predict the rate of change of the objective function. Cornet and Laroque (1987) establish conditions under which the values of the Lagrange multipliers are Lipschitz continuous for perturbations in the problem data.

Ganesh and Biegler (1987) have developed a sensitivity analysis based on the reduced Hessian. The reduction is conducted by using the equality constraints and the

implicit function theorem to reduce the dimensionality of the Hessian matrix that needs to be calculated. Their method is beneficial when there are equality constraints present in the formulation of the problem, because they have reduced the number of function evaluations required to find the required second order information numerically. Their method does not provide $\partial v/\partial p$ without calculating the full Hessian of the Lagrangian.

Rao (1987 a) and Guang-Yaun and Wen-Quan (1985) have studied the problem of dealing with fuzzy constraints and fuzzy objective functions. In their work they first solve a crisp problem then they attempt to calculate how far they can relax constraints while improving the objective function. To use their technique the user is required to specify how much violation is allowed in the constraints. Templeman (1987) reports using fuzzy set theory and optimization to design structures and deal with uncertainties in the problem.

Sandgren, Gim and Ragsdell (1985) describe a problem formulation that can be used to obtain optimum designs with a minimum sensitivity to uncontrollable parameters. Their approach does not use post optimality analysis but uses a modified objective function to deal with the uncertainties in the problem parameters.

The area of calculating sensitivity derivatives with respect to design variables (i. e. the calculation of gradients of functions) has been an area of active research. This can lead to significant savings over using finite differencing. The structural optimization community now widely uses sensitivity analysis when the finite element method is used to analyze a structure. An excellent survey article of methods of sensitivity analysis for structural optimization is provided by Adelman and Haftka (1986).

Haug and Arora, et al. (1977,1979,1981) have developed ways to calculate the gradients analytically for many structural and dynamic applications. Many of these methods are described in the book by Haug, Komkov and Choi (1985).

Sobieski, et al. (1981,1982,1983,1984,1985,1986,1987) has been working on developing sensitivity techniques for use with multi-level decomposition techniques. Decomposition methods break the solution of a large problem into a system level problem and a group of subproblems. Each subproblem is solved using a special formulation and inputs from the system level problem. A sensitivity analysis is performed on the subproblem and the results are feed as input to the system level problem. The system level problem gathers all the sensitivities of the subproblems and then based on these inputs and others, determines the next iteration of the process. Usually, the equations (2.6) - (2.9) are used at the subsystem level to determine the required sensitivities, but some difficulties have been encountered when changes in the active set occur.

Schmit and Chang (1984) have developed an extension of Sobieski's work and derived sensitivity equations for structural optimization problems. They derived more restrictive limits on the allowable perturbations than those provided by Sobieski. They have assumed that second derivatives of the constraints are available which is true of many structural problems but may not be true for other application areas.

Schmit and Chang formulated their structural optimization problem using reciprocal variables and solved for the sensitivity of the dual problem. For their structural problems, the Hessian of the Lagrangian was diagonally dominate and the Hessian of the objective function was analytically available. For this class of problems good results can be expected even if the Hessian of the Lagrangian is inaccurate.

Buyss and Gonin (1977) developed and implemented a sensitivity analysis procedure for an augmented Lagrangian (AL) type code, VF01A. Their implementation is encouraging because they make use of the approximations of the Hessian of the Lagrangian that were calculated during the solution of the original problem. The results that they obtained using the approximate matrices were in very close agreement of those obtained by using the exact matrices.

McKeown (1980 b) derives both the first and second order Kuhn-Tucker parameter sensitivity equations. He also provides a discussion of Fiacco's sensitivity for SUMT penalty functions versus Buyss and Gonin's sensitivity for AL penalty functions. He concludes that using sensitivity for AL penalty functions should be superior to sensitivity by SUMT because AL produces better conditioned matrices.

2.2. PREVIOUS WORK IN ESTIMATING PARAMETER SENSITIVITIES FOR CHANGES IN THE ACTIVE SET

When the active set of constraints changes, one of the underlying assumptions made in deriving the Kuhn-Tucker sensitivity equations is violated. This can result in inaccuracies in any extrapolations based on these sensitivities since, in general, a change in the active constraints will result in a different set of sensitivities. Accurate sensitivity analysis in the presence of active set changes is also very important for efficient convergence of the multi-level decomposition techniques proposed by Sobieski and, in general, for an accurate representation of the local sensitivities.

In the following subsections, we will first discuss the different cases that occur as a result of a constraint entering or leaving the active set, what effects these cases have on sensitivity analysis, and how changes in the active set can be predicted. We will then present examples of the sensitivities for the different cases which will also serve to indicate how the different sensitivity algorithms perform.

2.2.1. Cases to Consider

When a new constraint enters the active set, or a currently active constraint leaves the active set, we can expect a change in the sensitivity derivatives. However, it is also possible that the linear independence of the constraint gradients can also be affected. For the discussion that follows, we define the following four cases that can result from changes in the active set,

1. A constraint enters the active set and the constraint gradients are linearly independent.
2. A constraint leaves the active set and the constraint gradients are linearly independent.
3. A constraint enters the active set replacing an active constraint and the constraint gradients are linearly dependent.
4. A constraint enters the active set and feasible region disappears.

For Cases 1 and 2, we can expect discontinuities in the following derivatives when the active set changes: d^2f^*/dp^2 , $\partial x^*/\partial p$, and $\partial u^*/\partial p$.

Case 3 is characterized by a discontinuity in the Lagrange multiplier estimates which causes a discontinuity in df^*/dp . Since the active set changes there will also be a discontinuity in $\partial x^*/\partial p$. At the point where the constraints become linearly dependent, the Kuhn-Tucker sensitivity equations become singular. Often what is happening for Case 3 is that an exchange of constraints in the active set is about to take place (i.e. the new constraint may replace one of the constraints that is already in the active set). If the problem is not poorly formulated, we will find ourselves moving through the degenerate point as p increases or decreases and one of the constraints will be dropped from the active set.

Case 4 is characterized as a point from which p can only be perturbed in one direction. If p is perturbed in the wrong direction this will cause there to be no feasible region and there will be no solution for the optimization problem with this value of p . Thus we can only perturb p in the one direction that causes the optimum path to move into the feasible region, and there will only exist a directional derivative for the problem in that direction. Case 4 can be thought of as an overconstrained design where the designer adjusted a parameter to the point where the design is no longer able to meet specifications.

2.2.2. Prediction of when the Active set will Change

Barthelemy and Sobieski (1983 a) have observed that the accuracy of extrapolations of the objective function deteriorates rapidly when the active set changes. From section 2.1.2, we saw that we can use equations 2.10 and 2.11 to predict where the active set will change, thus we can use this information to predict when the extrapolations will deteriorate.

A problem with bounding Δp by equations 2.10 and 2.11 is that the estimate is only good for the first constraint that is encountered because once the active set changes the search direction to the new optimum will change (the discontinuity in $\partial x/\partial p$). Thus, it becomes very difficult to estimate when or which constraint will leave/enter the active set second. This problem will be addressed in section 6.

The merit of using equations 2.10 and 2.11 to predict when the active set will change was discussed by Adelman and Haftka (1986). They state, "The effectiveness of using this approach (equations 2.10 and 2.11) is still in doubt with positive results being obtained by Schmit and Chang (1984) and negative results being obtained by Barthelemy and Sobieski (1983 a)". We feel that the positive results that were obtained by Schmit and Chang are due to problem linearity and the changes in the active set that they encountered being case 1 and case 2 changes. We feel that the negative results obtained by Barthelemy and Sobieski are due to nonlinearity of the problem and also a case 3 change in the active set taking place. As we will see later in this report, the consequences on sensitivity derivatives of case 3 changes in the active set are often much more severe than case 1 and case 2 changes.

2.2.3. An Example of Case 1 and 2

The effect of a constraint entering or leaving the active set (Cases 1 and 2) can best be demonstrated by a simple example from Vanderplaats and Yoshida (1985).

$$\text{Minimize } f(x) = 2x_1^2 - 2x_1p + p^2 + 4x_1 - 4p \quad (2.13)$$

$$\text{subject to: } g_1 = 4p + x_1 \geq 0 \quad (2.14)$$

The Lagrangian will be

$$L(x,u) = 2x_1^2 - 2x_1p + p^2 + 4x_1 - 4p - u_1(4p + x_1) \quad (2.15)$$

for $p = 0$, the optimum is $f(x^*) = 0$, $x_1^* = 0$, $g_1 = 0$, and $u_1 = 4$.

This example will illustrate a constraint leaving the active set (case 2) as p increases. The same example can be used to illustrate a constraint entering the active set (case 1) if we use a different starting value of p .

To demonstrate the methods we have talked about, we will calculate the sensitivity estimates using four representative methods: the first and second order Kuhn-Tucker method, and first and second order extended design space method. We will conclude with a comparison of the various methods used to solve the problem.

To solve for the sensitivity by Kuhn-Tucker equations we use equation (2.6) to provide the following system of equations

$$\begin{bmatrix} 4 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial p} \\ \frac{\partial u_1}{\partial p} \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \end{bmatrix} \quad (2.16)$$

which yield

$$\frac{\partial x_1}{\partial p} = -4 \quad (2.17)$$

$$\frac{\partial u_1}{\partial p} = -18 \quad (2.18)$$

From equation (2.7) we can determine the sensitivity of the objective function with respect to the parameter p to be,

$$\frac{df}{dp} = \frac{\partial f}{\partial p} + \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial p} = -4 + 4(-4) = -20 \quad (2.19)$$

The active set will change when the Lagrange multiplier of the constraint goes to zero, which can be estimated by equation (2.10)

$$\Delta p = \frac{-u_1}{\left(\frac{\partial u_1}{\partial p}\right)} = \frac{-4}{-18} = 0.2222 \quad (2.20)$$

therefore we are assured of reasonable results for extrapolations for which Δp less than 0.2222.

For example, a linear approximation by equation (2.8) to estimate the value of the new optimum produce

$$f_{\text{new}} = f^* + \Delta p \frac{df}{dp} = 0 + \Delta p(-20) = -20\Delta p \quad (2.21)$$

A quadratic estimate of the new value of the objective function can be made by evaluating the following equation found in Fiacco (1983), McKeown (1980 b), and Sobieski and Barthelemy (1983)

$$\frac{d^2f}{dp^2} = \frac{\partial^2 L}{\partial p^2} + \frac{\partial^2 L}{\partial x_1 \partial p} \frac{\partial x_1}{\partial p} + \frac{\partial u_1}{\partial p} \frac{\partial g_1}{\partial p} \quad (2.22)$$

which produces $d^2f/dp^2 = 82$. Using the quadratic estimate for the value of the objective function we obtain

$$f_{\text{new}} = f^* + \Delta p \frac{df}{dp} + 0.5 \Delta p \frac{d^2f}{dp^2} \Delta p = -20\Delta p + 41\Delta p^2 \quad (2.23)$$

The same predictions can be made by Vanderplaats' extended design space algorithm. We begin by formulating the following direction finding problem for decreasing values of p , where x_2 represents the parameter p , and x_3 is an additional variable to ensure that p has the required sign.

$$\text{minimize } 4x_1 - 4x_2 - c x_3 \quad (2.24)$$

$$\text{subject to: } x_1 + 4x_2 \geq 0 \quad (2.25)$$

$$-x_2 - x_3 \geq 0 \quad (2.26)$$

$$1 - (x_1^2 + x_2^2) \geq 0 \quad (2.27)$$

For $c = 1000$, the solution is $x_1 = .970142$, $x_2 = -.242536$, $x_3 = .242536$ which yields the following estimates of the sensitivity derivatives

$$\frac{df}{dp} = -20 \quad (2.28)$$

$$\frac{dx_1}{dp} = -4 \quad (2.29)$$

For increasing values of p we obtain the following subproblem

$$\text{minimize } 4x_1 - 4x_2 - c x_3 \quad (2.30)$$

$$\text{subject to: } x_1 + 4x_2 \geq 0 \quad (2.31)$$

$$x_2 - x_3 \geq 0 \quad (2.32)$$

$$1 - (x_1^2 + x_2^2) \geq 0 \quad (2.33)$$

When this problem is solved, the resulting sensitivities are sensitive to the value of the parameter c . The solution for several values of c are presented below in Table 2.1.

Table 2.1 The effect of "c" on EDS sensitivity

variable	c=1000	c=500	c=100	c=10	c=1.0	c=0.0
x_1	-0.398E-2	-0.79E-2	-0.388E-1	-0.1763	-0.624	-0.707
x_2	0.99999	0.99996	0.99924	0.9611	0.9611	0.707
x_3	0.99999	0.99996	0.99924	0.9611	0.9611	0.707
df/dp	-4.016	-4.0316	-4.155	-5.150	-7.196	-8.0

From this table it is clear that the choice of c will effect the sensitivity derivatives. For demonstration purposes $c = 10$ was chosen, this yielded the following sensitivity derivatives.

$$\frac{df}{dp} = -5.1502 \quad (2.34)$$

$$\frac{dx_1}{dp} = -.28756 \quad (2.35)$$

Vanderplaats and Yoshida (1985) report that the value of c has little effect on the EDS algorithm. However Vanderplaats and Cai (1987) report that after further research the value of c will effect the accuracy of the EDS procedure.

Using Vanderplaats second order extended design space algorithm provides exact answers for the sensitivity for this problem.

Figure 2.1 illustrates the accuracy of various methods. We can see that when the active set changes at $\Delta p = 0.222$ the predictions become less accurate.

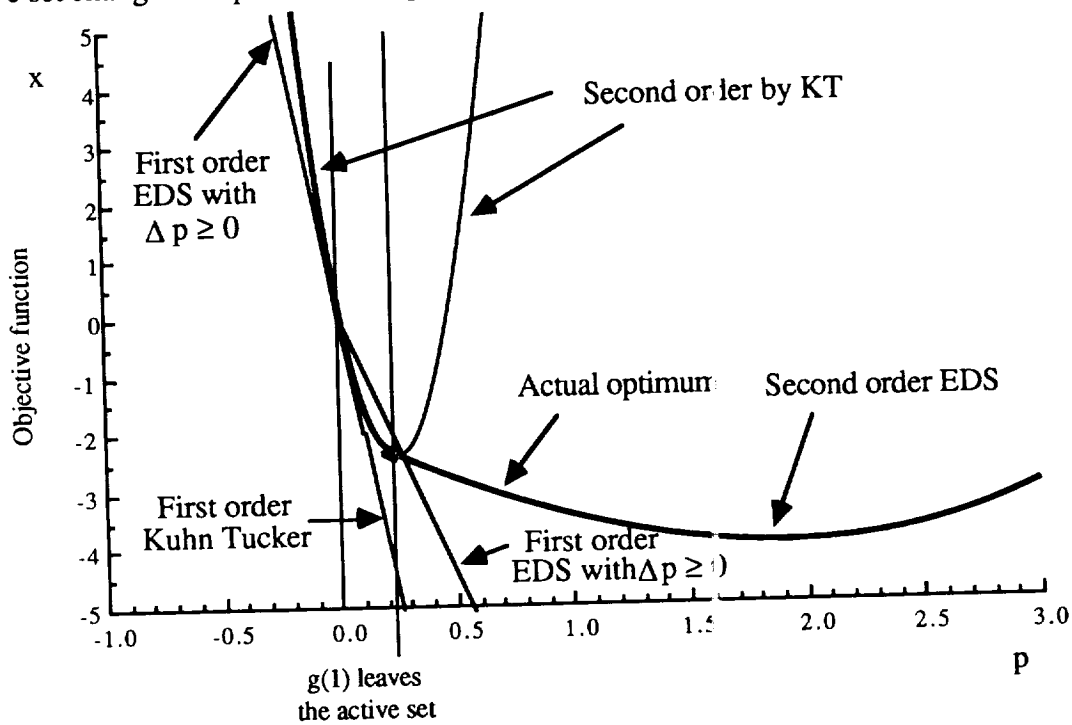


Figure 2.1 A plot of various optimal values of f with respect to p

Figure 2.2 illustrates the location of the optimum value of x_1 as a function of p , as predicted by various algorithms. When the active set changes there is a discontinuity in the rate of change of the optimum value of x_1 with respect to p (i. e. $\partial x_1 / \partial p$ is discontinuous at the point).

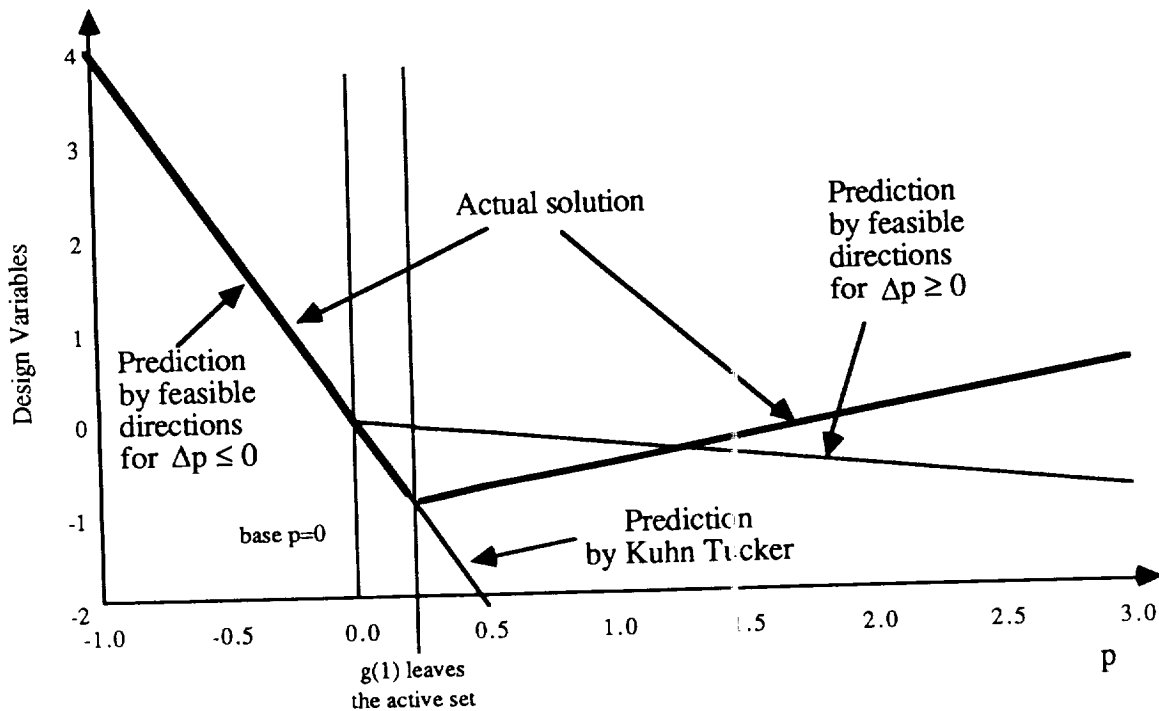


Figure 2.2 A plot of various optimal values of x_1 with respect to p

From figures 2.1 and 2.2 it is possible to draw some conclusions about the relative performance of the four different methods that were used to obtain sensitivity information. Using the first order Kuhn - Tucker method we see that the solution follows the inequality constraint in both the positive and negative direction. The linear estimate of the new value of x_1 is accurate for small changes in p less than 0.2222. But for values of p greater than 0.2222, the active set has changed and large errors in the predictions are introduced. This is also true for the linear prediction for the value of the objective function.

The second order Kuhn - Tucker estimate of the value of the objective function is in exact agreement in the region where the active set remains the same, as seen in figure 2.1. However after the active set changes the predicted value of the objective function is a poor predictor of the actual value of the optimum.

The first order extended design space provides the same results as the first order Kuhn-Tucker sensitivity for decreasing values of p . For increasing values of p we see that the search direction changes. This approximation appears to overcome the constraint leaving the active set, but it is a poor predictor of the actual value of the optimum for small variations in p . For other values of the parameter " c " we will obtain similar values for the sensitivity derivatives.

The second order extended design space provides the exact values of the locations of the optimum value of the objective function. This is because the approximating problem that is formulated is the same as the original problem.

With this simple example we have demonstrated the effect of a constraint leaving the active set on the algorithms for estimating parameter sensitivity. We can see from this example that, as we might anticipate, using second order estimates can produce more accurate extrapolations. In fact, only the second order extended design space algorithm provided good results after the constraint left the active set. However its usefulness is diminished by the need for second derivatives which can be computationally expensive to obtain.

2.2.4. Example of Case 3

Recall, that Case 3 is characterized by the adding of a new constraint to the active set and the gradients of the active constraints become linearly dependent. When the gradients of the constraints are linearly dependent the Lagrange multipliers will not be uniquely determined and the Kuhn-Tucker optimality conditions cannot be uniquely

verified. This also results in a discontinuity in the Lagrange multiplier sensitivities.

When the constraint gradients become linearly dependent for a value of one of the parameters it is assumed that this is only a temporary condition. If the user is interested in the effect of changing the parameter on the optimum then this information can be obtained on either side of the singular point.

This behavior is demonstrated in the following example

$$\text{minimize: } f = x_1^2 + (P - 1)^2 \quad (2.36)$$

$$\text{subject to: } g_1 = 3x_1 + 2P - 10 \geq 0 \quad (2.37)$$

$$g_2 = 2x_1 + 3P - 10 \geq 0 \quad (2.38)$$

When $P = 2$, the minimum $f^* = 5$ occurs at $x_1^* = 2$. At this point, both constraints are active, and the gradients of the constraints are not linearly independent. The Lagrange multipliers will be in the family

$$u_1, u_2 \in \{3u_1 + 2u_2 = 4, u_1 > 0, u_2 > 0\} \quad (3.39)$$

At this point, df^*/dp , $\partial x^*/\partial p$ and $\partial u/\partial p$ can not be uniquely determined. Results for these derivatives can be developed if we consider positive and negative changes in p separately on either side of this degenerate point which we shall indicate by $\partial x/\partial p^+$ for increasing values of p and $\partial x/\partial p^-$ for decreasing values of p .

Figure 2.3 presents the sensitivity plots for this problem. Figure 2.3 (a) and (b) represent the first order predictions of the new values of the Lagrange multipliers for this problem. For this problem the linear predictions agree with the optimum Lagrange multipliers. There is a discontinuity at $\Delta p = 0.0$, therefore there will only be directional derivatives for these values. Figure 2.3 (c) represents linear predictions of the new value of the objective function. Notice again that there is a discontinuity in the slope of the prediction and we can not determine df^*/dp for $\Delta p = 0$. Therefore df^*/dp will not exist for this value of p . Figure 2.3 (d) represents the predicted location of x_1 and we notice the same situation as we have for df^*/dp .

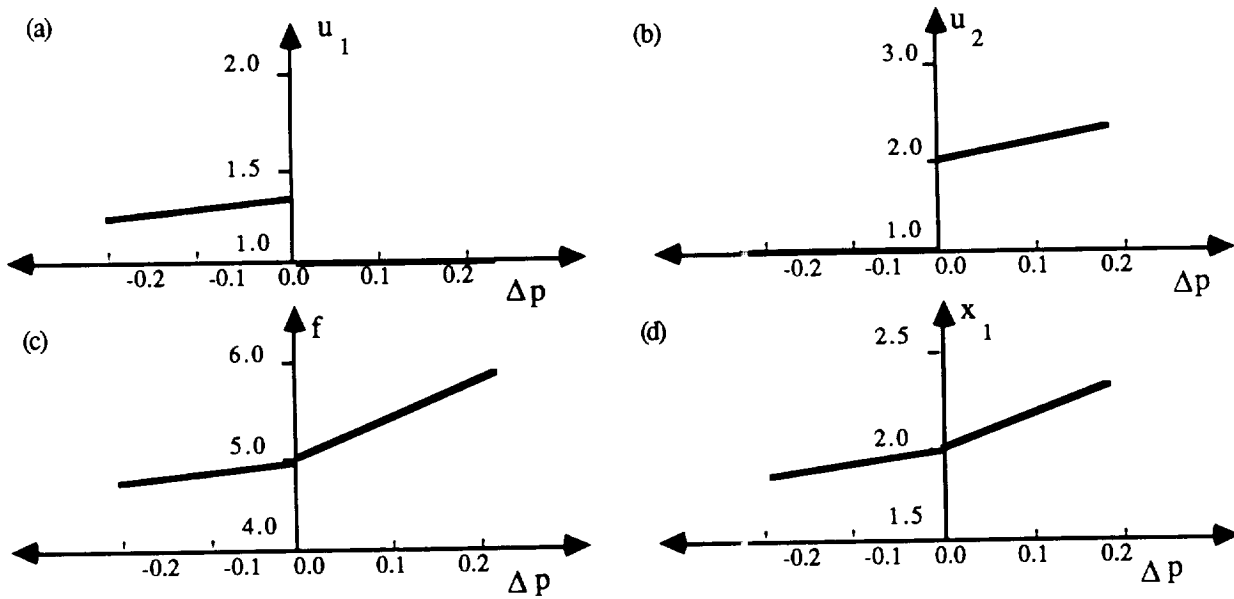


Figure 2.3. A comparison of the Sensitivity of a problem with a Linear Dependence in the Constraint normals

2.3. SUMMARY

Sensitivity analysis is now routinely used in linear programming (Falk and Fiacco 1982) and most linear programming algorithms provide modules for the calculation of sensitivities. This has not been the case for applications of nonlinear programming. The most common use of sensitivity derivatives has been in the area of structural optimization and in work done for decomposition methods. Some of the reasons for this may be due to a lack of understanding about how to perform sensitivity analysis for nonlinear problems, or to a lack of established procedures and supporting software that make the analysis more readily available to the average user. The largest contributor to its lack of use is probably the difficulty involved in implementing the current theory and methods.

An assessment of the methods discussed in Section 2.1 and demonstrated in the examples in Section 2.2.4 leads to the following conclusions about the current state of the art of parameter sensitivity analysis:

1. The Kuhn-Tucker sensitivity equations (2.6) accurately define the desired sensitivities assuming no changes in the active constraints. To implement these equations, however, requires second order information about the Hessian of the Lagrangian, and the change in the gradient of the Lagrangian with respect to the parameter. Both of which are difficult to obtain reliably for all but a few special cases.

2. The Extended Design Space (EDS) method provides sensitivity information without the need for the second order information required of the Kuhn-Tucker method. However, the sensitivity estimates are effected by a choice of a formulating parameter c , and may not give the same directions as those obtained from the Kuhn-Tucker method.
3. Changes in the active constraint set will effect the accuracy of any of the methods and may limit the region upon which extrapolations to the design can be relied upon.

3. New method for Estimating Parameter Sensitivity

In this chapter a new method for estimating parameter sensitivities based on the Recursive Quadratic Programming(RQP) method is described. We begin with a brief description of the RQP method and the advantages it provides for estimating sensitivities. Next, we present the RQP based algorithm for estimating parameter sensitivity that exploits the advantages of the RQP method discussed in the previous section. This is followed by a comparison of the new method with existing methods based on the type of information that is being produced and the number of function evaluations required. Finally, a discussion is presented of potential problems that may be encountered with the new RQP sensitivity method.

3.1. RQP METHODS

The RQP method has been on the forefront of recent research in optimization algorithms and has been emerging as one of the most efficient methods available for solving small to medium sized, general nonlinear programming problems (equations 1.1-1.6). State of the art RQP methods have been developed by many researchers, such as, Powell (1983), Schittkowski (1984), Gill, Murray and Wright (1986) and Bartholomew-Biggs (1986,1987) to name a few. The algorithm has been tested against other general nonlinear programming algorithms by Schittkowski (1980), Ecker and Kupferschmid (1984), Belegundu and Arora (1985). The results of these tests have shown the RQP method to be one of the most efficient algorithms available for the solution of nonlinear programming problems.

All RQP methods use the same basic strategy of linearizing the constraints and approximating the Hessian of the Lagrangian to form a quadratic programming (QP) subproblem. The QP subproblem is then solved for the search direction, s , and a new estimate the Lagrange multipliers of the constraints. The QP subproblem has the form

$$\text{Minimize } 1/2s^TBs + s^T\nabla f \quad (3.1)$$

$$\text{subject to } \nabla h^Ts + h = 0 \quad (3.2)$$

$$\nabla g^Ts + g \geq 0 \quad (3.3)$$

where B is an approximation to the Hessian of the Lagrangian which is normally constructed by variable metric methods. The Lagrange multipliers of the constraints for the original problem (equations 1.1-1.6) are estimated by the Lagrange multipliers of the constraints in the QP subproblem (equations 3.1-3.3). The search direction s is then used to calculate a new estimate of the optimum

$$\mathbf{x}_{it+1} = \mathbf{x}_{it} + \alpha \mathbf{s} \quad (3.4)$$

where α is determined by minimizing a line search penalty function P of the following general form,

$$P(\mathbf{x}, \mathbf{u}, \mathbf{v}, R) = f(\mathbf{x}) + R * \Omega(\mathbf{h}, \mathbf{g}, \mathbf{u}, \mathbf{v}) \quad (3.5)$$

where Ω represents some combination of the constraints and the Lagrange multipliers. The penalty function attempts to assure that both the objective function and the violation of the constraints are reduced. As the method converges, the optimal step length α generally approaches 1.

RQOPT, a typical RQP algorithm, was used in our research. A summary of the algorithm that is used by RQOPT is presented here, a full description of RQOPT can be found in the users manual (Beltracchi and Gabriele 1987 a), or Beltracchi (1985), Gabriele and Beltracchi (1986, 1987 b). There were several modifications that were made to RQOPT for this work and these will be discussed in section 4.1 of this report.

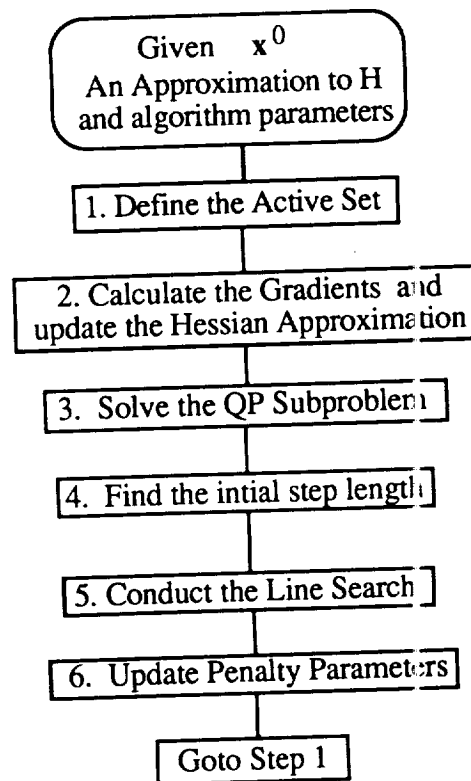


Figure 3.1 Flow Chart for RQOPT

Figure 3.1 shows the basic steps that are used by the RQOPT program. The RQOPT algorithm begins with an initial estimate of the location of the optimum and several algorithm parameters that have been set by the user. The first step of the algorithm is to identify the active constraints, it is important that the proper constraints are chosen to be in the active set as this can effect the rate of convergence of the algorithm and, for our

purposes, the approximation of the Hessian of the Lagrangian. Algorithm parameters are available to allow the user to control which constraints are considered active during the course of the optimization.

The second step is to calculate the gradients of the objective function and the constraints that are in the active set and then update the approximation of the Hessian of the Lagrangian. The update of the Hessian is performed using the BFS variable metric update with modifications specified by Powell (1977).

The third step is to solve a quadratic programming subproblem (equations 3.1-3.3). The QP subproblems generated by RQOPT are solved by OPTQP, a special implementation of the reduced gradient method. If the subproblem has no feasible solution, the active set is redefined by dropping constraints from the active set until a feasible subproblem can be found.

The line search for the next point x^{it+1} makes up the fourth and fifth steps of the algorithm. An initial step size for the line search is determined in the fourth step such that constraints not in the active set are not excessively violated. The line search is performed in the fifth step, and if a step of $\alpha = 1$ satisfies the line search criteria, then that step is taken and the line search ended.

The sixth step updates the penalty parameters used in the line search, and the Lagrange multiplier estimates. We then return to start another iteration.

There have been several different variants of the RQP method proposed. Some of the variants are discussed in Beltracchi (1985). The major differences in RQP algorithms are in the form of the line search objective function (equation 3.5) and the formulations of the QP subproblem (equations 3.1-3.3) that are used. Research continues on these areas but no one formulation has yet to prove itself clearly superior.

Some of the penalty functions that have been proposed for (3.5) are a l_1 exact penalty function (Fletcher 1984, Powell 1987), a l_2 quadratic loss penalty function (Bartholomew-Biggs 1980) or an augmented Lagrangian (Chen, Kong and Cha 1987, Bartholomew-Biggs 1985, 1987). The penalty function's parameters are adjusted after each iteration, and how the parameters are updated effects the convergence of the method.

There are two basic philosophies for forming the QP subproblem for RQP methods, the inequality constrained (IQP) formulation and equality constrained (EQP) formulation. The most common is the IQP approach which uses a subproblem of the form of equations 3.1-3.3. The EQP approach linearizes only a subset of the inequality

constraints and considers these as equality constraints in the subproblem (i.e. equation 3.3 is considered to be an equality constraint). A discussion of the advantages and disadvantages of the IQP and EQP subproblem formulation can be found in (Bartholomew-Biggs 1987, 1986, 1982, Zhou and Mayne 1985, Schittkowski 1983, Murray and Wright 1982, or Powell 1978).

Although the method does perform well, it does have some disadvantages. In general, the method produces a series of infeasible points while approaching the solution which may pose a problem for some problem formulations. RQP methods are also sensitive to variable and objective function scaling and no good scaling algorithms have been proposed. Finally, the best penalty function or algorithm for updating the penalty parameters for the line search is still a subject of a great deal of research in these methods.

On the plus side, the following advantages have been attributed to the method. In terms of number of function evaluations, this method appears to be one of the most efficient methods available. This has been demonstrated in any of the published comparison studies in which codes for these methods participate. The method does not require a feasible starting point which means there is no special phase 1 search employed as in the GRG method or the feasible direction method. Although, as mentioned above, the method is sensitive to variable and objective function scaling, it is not sensitive to constraint scaling. Finally, the RQP method provides an estimate of the Hessian of the Lagrangian, which can be useful for other purposes, and it is very efficient at locating an optimum when the starting point is close to the true optimum. Both of these last advantages will be exploited in the next section which describes a method for sensitivity estimation based on the RQP method.

3.2. PROPOSED ALGORITHM FOR PARAMETER SENSITIVITY

In reviewing the current methods for sensitivity analysis in chapter 2, we recall that to employ the Kuhn-Tucker sensitivity equations required second order information about the Lagrangian. For most engineering problems this type of information is often not available in closed form, and estimation techniques would be prone to truncation and numerical errors. Therefore, the application of these equations to a broad spectrum of engineering applications is limited.

One proposal mentioned in chapter 2 to circumvent these problems was suggested by Armacost and Fiacco (1974) and McKeown (1980 b). Their proposal to estimate the sensitivities without estimating the higher order information was given in equations 2.4 and 2.5,

$$\frac{df^*}{dp} = \frac{f(x^*, p + \Delta p) - f(x^*, p - \Delta p)}{2 \Delta p}$$

$$\frac{\partial x^*}{\partial p} = \frac{x^*(p + \Delta p) - x^*(p - \Delta p)}{2 \Delta p}$$

These equations represent the use of differencing techniques to estimate the sensitivities, where the values $f(x^*, p + \Delta p)$, $x^*(p + \Delta p)$, etc. are determined by reoptimizing the problem for the new values of the parameter. For most algorithms, particularly penalty function based methods, the reoptimizations would be a non-trivial task requiring a considerable number of function evaluations. However, this is the type of problem where the RQP method is considered to be very effective. The goal of the new algorithm is to exploit the strengths of the RQP method to estimate sensitivities by these differencing techniques.

The RQP method possesses two characteristics that we felt can be exploited for determining parameter sensitivities: (1) an approximation to the Hessian of the Lagrangian is developed, and (2) if this approximation is exact (or close) then the RQP method quickly and efficiently solve the reoptimization problem used in the difference equations. Essentially, if we can develop good Hessian approximations, the RQP method is equivalent to applying Newton's method to solve the Kuhn-Tucker conditions for the perturbed problems which should require only 1 or 2 iterations of RQP¹. The small number of iterations, coupled with the fact that the RQP method should require only a one step line search, should allow the reoptimizations to occur without the need for many function evaluations.

Based on the above arguments, we propose the following procedure to calculate parameter sensitivity derivatives (for cases where there are no changes in the active set for small variations in the parameters²).

Step 0. Given an optimal solution x^* , f^* , u^* , an active set of constraints, and an approximation to the Hessian of the Lagrangian, all achieved by convergence of the RQP method.
(the * notation is used to denote optimum values)

Step 1. Perturb the fixed parameter p_i to $p_i^+ = p_i^0 + \Delta p_i$ where Δp_i is some small perturbation to p_i

Step 2. Perform one complete iteration of the RQP method to find:
 f^+ the estimated value of the optimum objective function
 x^+ the estimated value of the optimal of the design variables

¹ We can expect only one or two iterations of RQP if we can adequately approximate the perturbed problem with a quadratic function. Due to the small region of interest, a quadratic approximation should be good.

² At points where the active set changes then modifications discussed in chapter 6 must be used to calculate directional derivatives.

u^+ the estimated value of the optimum Lagrange multipliers

$g_j^+ \quad j \notin \text{Active set}$
(as predicted by the RQP method for $p_i = p_i^-$)

Step 3. Perturb the fixed parameter p_i to $p_i^- = p_i^0 - \Delta p_i$

Step 4. Perform one complete iteration of the RQP method to find:

f^- the estimated value of the optimum objective function

x^- the estimated value of the optimal of the design variables

u^- the estimated value of the optimum Lagrange multipliers

$g_j^- \quad j \notin \text{Active set}$

(as predicted by the RQP method for $p_i = p_i^-$)

Step 5. Obtain estimates for the sensitivity derivatives from the following central difference approximations

$$\frac{df^*}{dp} = \frac{f^+ - f^-}{2\Delta p} \quad (3.6)$$

$$\frac{\partial x^*}{\partial p} = \frac{x^+ - x^-}{2\Delta p} \quad (3.7)$$

$$\frac{\partial u^*}{\partial p} = \frac{u^+ - u^-}{2\Delta p} \quad (3.8)$$

Step 6. Estimate the sensitivity of the inactive constraints by

$$\frac{dg_j^*}{dp} = \frac{g_j^+ - g_j^-}{2\Delta p} \quad j \notin \text{Active set} \quad (3.9)$$

In addition to the algorithm described above, the following variants of the basic algorithm are also proposed

1. Forward differencing. For this variant we would omit steps 3 and 4 and then use a forward difference approximation (equation 3.10 instead of equations 3.6-3.9) to approximate the derivatives

$$\frac{\partial q^*}{\partial p} = \frac{q^+ - q}{\Delta p} \quad (3.10)$$

where q can represent f^* , x , u , and the inactive constraints. We may want to use this formulation because it requires less function evaluations than the central difference approximation. However, the forward difference approximation is more susceptible to roundoff and truncation errors and requires a more accurate optimum to yield good sensitivity derivatives.

2. Forward differencing using 2 iterations of the RQP method. This variant is

similar to variant 1, but we would perform 2 iterations of RQOPT in step 2. This will yield a more accurate estimate of the optimum of the perturbed problem. When we use this option we can also update the approximation to the Hessian of the Lagrangian, or adjust the perturbation Δp to obtain a more accurate estimate of the derivatives.

3. Central differencing using 2 iterations of the RQP method. This variant would perform two iterations of RQOPT in steps 2 and 4 of the basic algorithm. As in variant 2 we can update the Hessian approximation during each iteration or adjust the perturbation Δp to obtain a more accurate estimate of the derivatives. This variant is the most computationally expensive of the proposed variants.

When there are many parameters that the user needs to obtain sensitivities for then the user may want to use variant 2 or variant 3 to calculate the sensitivities for the first few parameters. This will allow a more accurate estimate of the Hessian of the Lagrangian to be constructed. After an accurate estimate of the Hessian of the Lagrangian is built, the user should switch to either the baseline or variant 1 to obtain the sensitivities of the remaining parameters. The Kuhn-Tucker sensitivity equations may also be used with the Hessian approximation, after a good estimate of the Hessian of the Lagrangian is built. However the Kuhn-Tucker sensitivity equations also require $\partial \nabla_x L / \partial p$ be calculated and this term may be subject to numerical noise because $\nabla_x L \approx 0$.

3.3. COMPARISON TO EXISTING METHODS

This section provides a derivation that indicates the performance that is expected from the new sensitivity algorithm. This section also presents a comparison between the RQP based method and two existing methods described in chapter 2 based on the number of function evaluations required to estimate the sensitivities.

3.3.1. Demonstration of Equivalence of New Method to Kuhn-Tucker Method

This section will show that the finite difference approximations obtained by the proposed method are in fact equivalent to the sensitivities obtained by solving a modified set of Kuhn-Tucker sensitivity equations. The modification of the Kuhn-Tucker sensitivity equations involves replacing the Hessian of the Lagrangian with the approximation B , obtained from the RQP method.

The following assumptions are made for this derivation; no equality constraints are present, the base optimal point is stable³, and the gradients are continuous. The derivation

³ A stable point is defined as a point where the active set does not change for small variations in the parameters

in the presence of equality constraints does not change too much but the equality constraints were left out to simplify the notation. If the base point is not stable then this derivation can be used to find directional derivatives; this will be discussed at the end of this section. If the gradients are not continuous then we cannot even be assured of an optimum point since the assumption of continuity is also made for the derivation by the Kuhn-Tucker method.

We begin by restating the Kuhn-Tucker Sensitivity equations

$$\begin{bmatrix} \nabla_x^2 L & -\nabla_x g \\ \nabla_x g^T & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial p} \\ \frac{\partial u}{\partial p} \end{bmatrix} + \begin{bmatrix} \frac{\partial \nabla_x L}{\partial p} \\ \frac{\partial g}{\partial p} \end{bmatrix} = 0 \quad (3.11)$$

We strive in this derivation to show that the proposed method is equivalent to estimating the sensitivities using modified version of equation (3.11) that replaces $\nabla_x^2 L$ with **B** obtained

from the RQP method. If this is the case, then we can anticipate the kind of accuracy to expect and where the possible sources of error will result.

If we examine the equations (3.6-3.8), used by the proposed RQP sensitivity method we see that these provide finite difference approximations to the sensitivity derivatives of the objective function, design variables, and Lagrange multipliers with respect to p_i . The derivatives are defined by the following

$$\frac{df^*}{dp} = \lim_{\Delta p \rightarrow 0} \left(\frac{f^*(x^* + \Delta x, p^0 + \Delta p) - f^*(x^*, p^0)}{\Delta p} \right) \quad (3.12)$$

$$\frac{\partial x^*}{\partial p} = \lim_{\Delta p \rightarrow 0} \left(\frac{x^*(p^0 + \Delta p) - x^*(p^0)}{\Delta p} \right) \quad (3.13)$$

$$\Delta x = \frac{\partial x^*}{\partial p} \Delta p \quad (3.14)$$

$$\frac{\partial u^*}{\partial p} = \lim_{\Delta p \rightarrow 0} \left(\frac{u^*(p^0 + \Delta p) - u^*(p^0)}{\Delta p} \right) \quad (3.15)$$

where p^0 represents our base point.

The RQP subproblem for the simplified case where the active constraints remain active and there are no equality constraints can be written as

$$\min 1/2 s^T B s + s^T \nabla_x f \quad (3.16)$$

$$\text{subject to } s^T \nabla_x g_j + g_j = 0 \quad j \in \text{Active Set} \quad (3.17)$$

where **B** is the approximation to the Hessian of the Lagrangian and the inequality

constraints g_j are considered as equality constraints.

If we assume⁴ a step length of $\alpha = 1$ is used in the line search (equation 3.4) we can rewrite equation 3.4 in terms of $\mathbf{x}' - \mathbf{x}$ as

$$\mathbf{s} = \mathbf{x}' - \mathbf{x} \quad (3.18)$$

where \mathbf{x}' is the new estimate of \mathbf{x}^* . Substituting equation 3.18 into equations 3.16 and 3.17 we obtain the following subproblem which is minimized with $(\mathbf{x}' - \mathbf{x})$ as the design variables

$$\min 1/2 (\mathbf{x}' - \mathbf{x})^T \mathbf{B}(\mathbf{x}' - \mathbf{x}) + (\mathbf{x}' - \mathbf{x})^T \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}_i) \quad (3.19)$$

$$\text{subject to } (\mathbf{x}' - \mathbf{x})^T \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}_i) + g_j(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}_i) = 0 \quad j \in \text{Active Set} \quad (3.20)$$

We can now state the optimality conditions for the subproblem represented in equations (3.19-20) as

$$\mathbf{B}(\mathbf{x}' - \mathbf{x}) + \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}_i) - \mathbf{u}' \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}_i) = 0 \quad (3.21)$$

$$(\mathbf{x}' - \mathbf{x})^T \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}_i) + g_j(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}_i) = 0 \quad j \in \text{Active Set} \quad (3.22)$$

Here \mathbf{u}' represents the estimated value of the Lagrange multipliers at the new optimum.

Now we substitute into equation (3.21) the following definitions of zero

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{p}^0) = 0 = \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{p}^0) - \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0) \quad (3.23)$$

$$\mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) - \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) = 0 \quad (3.24)$$

This will yield

$$\begin{aligned} & \mathbf{B}(\mathbf{x}' - \mathbf{x}) + \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) - \mathbf{u}' \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) - (\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{p}^0) - \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0)) + \\ & \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) - \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) = 0 \end{aligned} \quad (3.25)$$

Rearranging we obtain

$$\begin{aligned} & \mathbf{B}(\mathbf{x}' - \mathbf{x}) - \mathbf{u}' \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) + \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) + \\ & (\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) - \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p})) - (\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{p}^0) - \mathbf{u} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{p}^0)) = 0 \end{aligned} \quad (3.26)$$

Rearranging further and writing in terms of the Lagrangian function we obtain

$$\mathbf{B}(\mathbf{x}' - \mathbf{x}) - (\mathbf{u}' - \mathbf{u}) \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) + \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0 + \Delta \mathbf{p}) - \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0) = 0 \quad (3.27)$$

Now we will divide equation (3.27) by $\Delta \mathbf{p}$ and take the limit as $\Delta \mathbf{p}$ goes to zero to

⁴ A common assumption for RQP methods

obtain

$$\lim_{\Delta p \rightarrow 0} \frac{\mathbf{B}(\mathbf{x}' - \mathbf{x}) - (\mathbf{u}' - \mathbf{u}) \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p) + \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0 + \Delta p) - \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0)}{\Delta p} = 0 \quad (3.28)$$

Using the additive and multiplicative properties of the Limit function we obtain

$$\begin{aligned} \mathbf{B} \lim_{\Delta p \rightarrow 0} \left(\frac{\mathbf{x}' - \mathbf{x}}{\Delta p} \right) - \lim_{\Delta p \rightarrow 0} \left(\frac{\mathbf{u}' - \mathbf{u}}{\Delta p} \right) \lim_{\Delta p \rightarrow 0} (\nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p)) + \\ \lim_{\Delta p \rightarrow 0} \left(\frac{\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0 + \Delta p) - \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0)}{\Delta p} \right) = 0 \end{aligned} \quad (3.29)$$

Now we can use the definition of a derivative of some function h with respect to some variable p

$$\frac{\partial h}{\partial p} = \lim_{\Delta p \rightarrow 0} \frac{h(p + \Delta p) - h(p)}{\Delta p} \quad (3.30)$$

Applying the definition of $\partial \mathbf{x} / \partial p$, $\partial \mathbf{u} / \partial p$ to (3.29) we obtain

$$\mathbf{B} \frac{\partial \mathbf{x}}{\partial p} - \frac{\partial \mathbf{u}}{\partial p} \lim_{\Delta p \rightarrow 0} \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p) + \frac{\partial \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0)}{\partial p} = 0 \quad (3.31)$$

If we use the standard assumption that the functions are twice continuously differentiable we can state

$$\lim_{\Delta p \rightarrow 0} \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p) = \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0) \quad (3.32)$$

And now substituting equation (3.32) into equation (3.31) we obtain

$$\mathbf{B} \frac{\partial \mathbf{x}}{\partial p} - \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0) \frac{\partial \mathbf{u}}{\partial p} + \frac{\partial \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}, \mathbf{p}^0)}{\partial p} = 0 \quad (3.33)$$

The equation above (equation 3.33) represents the first part of the Kuhn-Tucker sensitivity equations with the approximation \mathbf{B} instead of the Hessian of the Lagrangian.

The next step in this derivation is to examine equation (3.22) in terms of $\mathbf{p}^0 + \Delta p$ we obtain

$$(\mathbf{x}' - \mathbf{x})^T \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p) + g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p) = 0 \quad j \in \text{Active Set} \quad (3.34)$$

Now we can subtract $g_j(\mathbf{x}, \mathbf{p}) = 0$ from equation (3.34) to obtain

$$(\mathbf{x}' - \mathbf{x})^T \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p) + g_j(\mathbf{x}, \mathbf{p}^0 + \Delta p) - g_j(\mathbf{x}, \mathbf{p}^0) = 0 \quad j \in \text{Active Set} \quad (3.35)$$

If we divide equation (3.35) by Δp and take the limit as Δp goes to zero we can write

$$\lim_{\Delta p \rightarrow 0} \left(\frac{(\mathbf{x}' - \mathbf{x})^T \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p})}{\Delta p} + \frac{g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) - g_j(\mathbf{x}, \mathbf{p}^0)}{\Delta p} \right) = 0 \quad (3.36)$$

Using the additive and multiplicative properties of the limit function we obtain

$$\lim_{\Delta p \rightarrow 0} \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) * \lim_{\Delta p \rightarrow 0} \left(\frac{\mathbf{x}' - \mathbf{x}}{\Delta p} \right) + \lim_{\Delta p \rightarrow 0} \left(\frac{g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) - g_j(\mathbf{x}, \mathbf{p}^0)}{\Delta p} \right) = 0 \quad (3.37)$$

Again using the definition of a partial derivative of (equation 3.30) and we obtain from equation (3.37)

$$\lim_{\Delta p \rightarrow 0} \nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0 + \Delta \mathbf{p}) * \frac{\partial \mathbf{x}}{\partial p} + \frac{\partial g_j}{\partial p} = 0 \quad (3.38)$$

Using the results in equation (3.32) we obtain

$$\nabla_{\mathbf{x}} g_j(\mathbf{x}, \mathbf{p}^0) * \frac{\partial \mathbf{x}}{\partial p} + \frac{\partial g_j}{\partial p} = 0 \quad (3.39)$$

Which represents the second part of the Kuhn-Tucker sensitivity equations.

Now equations (3.33) and (3.39) can be assembled into matrix form to yield

$$\begin{bmatrix} \mathbf{B} & -\nabla_{\mathbf{x}} \mathbf{g} \\ \nabla_{\mathbf{x}} \mathbf{g}^T & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial p} \\ \frac{\partial \mathbf{u}}{\partial p} \end{bmatrix} + \begin{bmatrix} \frac{\partial \nabla_{\mathbf{x}} L}{\partial p} \\ \frac{\partial \mathbf{g}}{\partial p} \end{bmatrix} = 0 \quad (3.40)$$

Equation 3.40 is the same as equation 3.19 with the exception that equation 3.40 uses the approximation, \mathbf{B} , of the Hessian of the Lagrangian in place of, $\nabla_{\mathbf{x}}^2 L$, the true Hessian of the Lagrangian. Referring to (3.40) as the modified Kuhn-Tucker equations, we see that the proposed method is principally a difference approximation to the modified Kuhn-Tucker equations. This implies that if \mathbf{B} is a good approximation of the Hessian of the Lagrangian, and a proper choice can be made for the difference parameter that minimizes truncation and roundoff errors, then we can produce sensitivity derivatives without the need to obtain or estimate the second derivatives required of the Kuhn-Tucker method.

Several examples were tested to see if the sensitivity derivatives estimated by the RQP method with one iteration converged to the value of sensitivity derivatives estimated by the Kuhn-Tucker sensitivity with the approximate Hessian. From these examples we observed that the sensitivity derivatives delivered by the new RQP algorithm are close to the derivatives approximated by the Kuhn-Tucker method with the Hessian approximation. One of these examples is presented here to show this agreement.

Test problem 2 (which is described in the appendix) is used to demonstrate the equivalence of the new method to the Kuhn-Tucker method. The starting point of $\mathbf{x}^0 = (1.1, 1.1, 1.1)$ was used. RQOPT (with the BFS update and $\mathbf{H}^0 = \mathbf{I}$) solved the problem in one iteration and yielded the following approximation to the Hessian matrix⁵

$$\mathbf{H}_{\text{approx}} = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{bmatrix}$$

If we use this Hessian approximation to solve for the sensitivity of parameter 1 by equation (3.40) we will obtain the following system of equations

$$\begin{bmatrix} 3 & 2 & 2 & -1 \\ 2 & 3 & 2 & -1 \\ 2 & 2 & 3 & -1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial p_1} \\ \frac{\partial x_2}{\partial p_1} \\ \frac{\partial x_3}{\partial p_1} \\ \frac{\partial u_1}{\partial p_1} \end{bmatrix} + \begin{bmatrix} -12 \\ 5 \\ 0 \\ 1 \end{bmatrix} = 0 \quad (3.41)$$

the solution of these equations yields

$$\frac{\partial \mathbf{x}}{\partial p_1} = (9.33333, -7.66666, -2.66666) \quad (3.42)$$

$$\frac{\partial u}{\partial p_1} = (-4.66666) \quad (3.43)$$

The RQP based sensitivity algorithm calculated the following sensitivity derivative approximations.

$$\frac{\partial \mathbf{x}}{\partial p_1} = (9.33333, -7.66666, -2.66666) \quad (3.44)$$

$$\frac{\partial u_1}{\partial p_1} = (-4.66666) \quad (3.45)$$

The above derivatives were calculated using the RQSEN program (described in section 4 and the appendix of this report) with a perturbation of $\Delta p = 0.0001$ (using central differencing, equations 3.7, 3.8) and one iteration of RQP to solve the perturbed problems.

If the base point, \mathbf{p}^0 , is unstable (degenerate) we can use a similar derivation to calculate directional derivatives, which will be useful for predicting the sensitivities of the design variables and Lagrange multipliers. The use of directional derivatives will be discussed in section 6.

⁵The Hessian approximation for problem 2 is not close to the true Hessian of the Lagrangian (given in the appendix of this report). This is because the starting point was chosen to produce a poor approximation so we could clearly indicate the performance of the RQP sensitivity method in comparison to the Kuhn-Tucker sensitivity method with the approximate Hessian from RQOPT

3.3.2. Performance Comparison with Other Methods

This section compares the RQP based method to two of the methods discussed in chapter 2; the Kuhn-Tucker method, and the extended design space (EDS) method. The comparison is based on Table 3.1 which examines the number of function evaluations required by each method to calculate parameter sensitivities df^*/dp , $\partial x^*/\partial p$ and $\partial u^*/\partial p$ (assuming that when the optimum is found that the Kuhn-Tucker conditions have been checked, this means that $\nabla_x L$ and the Lagrange multipliers are known before the sensitivity analysis is performed). It is assumed that the objective function and constraints are interrelated⁶. It is also assumed that problem linearity or problem form are not exploited in calculating parameter sensitivities.

The first row of Table 3.1 represents the methods used in this comparison. The second row represents the number of function evaluations required to calculate the sensitivity derivatives for the first parameter. Subsequent parameters may require fewer evaluations for some methods.

The first column of Table 3.1 represents the number of variables present in the problem. The second column represents the amount of work required to solve for the sensitivities using the Kuhn-Tucker sensitivity equations. The third and fourth columns represent the number of function evaluations required by the EDS algorithm. Column 3 represents the first order method and column 4 the second order method. The fourth column, RQP 1, indicates that forward difference approximations were used to calculate the gradients. The fifth column RQP 2, also uses forward difference approximations but 2 iterations of RQOPT are allowed during the reoptimization. The fifth column RQP 3 represents the amount of work required for the base line algorithm using central difference approximations. The sixth column RQP 4 represents using central difference approximations with 2 iterations of RQOPT.

If the objective function sensitivity is calculated by equation 2.12 ($df^*/dp = \partial f/\partial p - u \partial g/\partial p$) then assuming that objective and constraint information can both be obtain in one call, only one extra function evaluation is required to determine $\partial f/\partial p$ and $\partial g/\partial p$. However, if one wants the design variable and Lagrange multiplier sensitivity then some other equations must be used.

⁶The value of the objective function and all of the constraints are calculated by one subroutine

n	Kuhn-Tucker	EDS (1 st)	EDS(2 nd)	RQP 1	RQP 2	RQP 3	RQP 4
	$\frac{n^2}{2} + \frac{3n}{2} + 1$	1	$\frac{(n+1)^2}{2} + \frac{3(n+1)}{2}$	n+1	2n+2	2n+1	4n+2
1	4	1	5	2	4	3	6
2	6	1	9	3	6	5	10
3	10	1	14	4	8	7	14
4	15	1	20	5	10	9	18
5	21	1	27	6	12	11	22
10	66	1	77	11	22	21	42
15	136	1	152	16	32	31	62
20	231	1	252	21	42	41	82
40	861	1	902	41	82	81	162

Note: RQP 1 uses forward difference approximations and one iteration to solve the perturbed problem
RQP 2 uses forward difference approximations and two iterations to solve the perturbed problem
RQP 3 uses central difference approximations and one iteration to solve the perturbed problem
RQP 4 uses central difference approximations and two iterations to solve the perturbed problem

Table 3.1 Comparison of Various Algorithms for use in sensitivity analysis

The following observations can be drawn from this table.

1. For the Kuhn-Tucker sensitivity equations, most of the work in finding the parameter sensitivity is involved in the calculation (by finite differences) of the Hessian of the Lagrangian. However, after the first parameter sensitivity is determined the cost of evaluating successive sensitivity derivatives is reduced to (n+1) extra function evaluations.
2. For the first order EDS algorithm, the work required to calculate the parameter sensitivity does not increase with problem size. However, this algorithm will not deliver $\partial u/\partial p$ and this algorithm may not be able to find the correct value for $\partial x/\partial p$. This will mean that df^*/dp will also be inaccurate with this method. If the problem is fully constrained the accuracy of $\partial x/\partial p$ is better but the method may still provide inaccurate derivatives.
3. For the second order EDS algorithm most of the work is in the calculation of the Hessian of the objective function and the Hessian of the constraints. The work involved for calculation of successive parameter sensitivities only requires approximately n+2 extra function evaluations. This algorithm requires the solution of a quadratic approximating problem for every new value of the parameter supplied by the user.
4. For the RQP 1 algorithm (forward differencing) is the most efficient of the RQP methods proposed and seems to be much more efficient than the Kuhn-Tucker algorithm. The work required to calculate successive parameter derivatives is constant (n+1 function evaluations). This algorithm will perform well when B is a good approximation and the perturbation Δp is properly chosen.
5. The number of function evaluations for the RQP 2 algorithm (forward differencing and 2 iterations of the RQOPT algorithm) grows linearly. The work required to calculate successive parameter derivatives is constant (2n+2 function evaluations). The work for calculating successive parameter sensitivities may be reduced because the Hessian approximation will improve after each parameter sensitivity derivative is approximated, which will eventually reduce the amount

of work required to solve the perturbed problem.

6. For the RQP 3 algorithm (central differencing) the work involved grows linearly and the work for calculating successive parameter derivatives is constant ($2n+2$ function evaluations). An indication of nonlinearity of the sensitivity derivatives can be indicated by checking for second derivatives of the functions as follows

$$\frac{d^2f}{dp^2} \approx \frac{f^+ - 2f^* + f^-}{\Delta p^2} \quad (3.46)$$

This approximation of the second derivatives may not yield accurate results but it may be able to indicate that there is curvature present in the problem. Another advantage of using central differences occurs when the active set changes and directional derivatives can be approximated.

7. The RQP algorithm with central differencing and 2 iterations of RQOPT is the most expensive of the proposed RQP algorithms. The work required to calculate successive parameter derivatives is constant ($4n+2$ function evaluations). The work for calculating successive parameter sensitivities will be reduced if we allow updating of the Hessian approximation during the RQP iterations, as less work will be required to solve the perturbed problem when the Hessian approximation is improved.

The above discussion dealt with the number of required function evaluations to calculate the parameter sensitivities. We did not account for any of the other overhead such as solving the QP subproblem for the RQP method or solving a quadratic approximating problem for the second order EDS algorithm.

The overhead associated with using the Kuhn-Tucker sensitivity equations is relatively small after the first parameter sensitivity is calculated, this is because if a factorization (i.e. LU) is used to solve the Kuhn-Tucker sensitivity equations then the amount of overhead becomes $o(n)$ flops. The overhead for solving the RQP subproblems will also be relatively small if a good implementation of the RQP method is used (i.e. a procedure proposed by Gill et. al. (1987) requires only $o(n)$ flops). The overhead for the first order EDS method will also be relatively small. However the overhead for the second order EDS method could be large depending on the problem.

In summary, the RQP based methods are competitive with the existing methods. All variants of the RQP based method require approximately the same number of function evaluations for small problems ($n < 5$), but considerably less for larger problems ($n > 5$).

3.4. POTENTIAL PROBLEMS

One of the main issues that needs to be investigated concerns the Hessian approximation: will the approximation converge in practice as predicted by the theory? If

convergence has not taken place then we need to investigate how to improve the Hessian approximation. Some modifications that can be made to obtain a more accurate Hessian approximation are discussed in Chapters 4 and 5.

As with the estimation of any gradient by finite differences, the perturbation step size Δp and the nonlinearity of the problem will effect accuracy of the derivative approximation. Rules from Gill, Murray and Wright (1983) or Adelman, Haftka, and Iott (1986) can be investigated as a means to select the step size Δp . An automated selection procedure for Δp should be investigated after the initial RQP sensitivity algorithm is tested.

When using the forward difference option the choice of Δp is even more critical. If Δp is too small and the optimum of the problem is not known exactly then when the perturbed problem is solved we may only be seeing a better estimate of x^* being found rather than an estimate of the solution of the perturbed problem. This will cause the derivative approximations to be inaccurate. If Δp is too large then we may only be obtaining trend information for the problem.

All optimization programs incorporate some kind of convergence criteria that is based on the relative change in the design variables. This stopping criteria will effect the calculation of the sensitivity derivatives for all available methods, because there is a common assumption that the base point is a true optimum. The central difference approximation may be less sensitive to inexact solutions because the solution of the perturbed problems will be of a similar degree of accuracy.

When solving the quadratic programming subproblem some type of convergence criteria is normally used. How small this tolerance is will effect how much work is needed to solve the subproblem (Nash 1985). During the early stages of the optimization it is not advisable to locate the exact solution of the QP subproblem as this may be too expensive. However once the program is in the region of a minimum the solution of the subproblem needs to be accurate. Therefore, we expect to use a tight convergence criteria for our QP solver during our reoptimizations.

3.5. SUMMARY

We have proposed a method and some variants based on the RQP method for estimating parameter sensitivities which provides sensitivity estimates nearly equivalent to the Kuhn-Tucker method. The method avoids the need for calculating second derivatives and its efficiency is competitive with current methods. The accuracy of the method

depends on two major pieces of information, the quality of the Hessian approximation provided by the RQP method, and the step size of the difference parameter used in the difference formula. Both these aspects of the method will be discussed in the following chapters.

4. Implementation

This chapter will discuss the implementation of the new parameter sensitivity method described in chapter 3. The program used as the basis for testing the new method was the RQOPT program which is an implementation of an active set RQP method (Beltracchi and Gabriele, 1987). The discussion begins with a discussion of the modifications made to RQOPT to perform the necessary calculations, and ends with a description of the software system developed to calculate parameter sensitivities.

4.1. MODIFICATIONS TO RQOPT

Most of the modifications to RQOPT were concentrated in one of the major areas of concern for the new sensitivity algorithm, the Hessian approximation. These modifications are discussed in subsections 4.1.1 and 4.1.2. The line search of RQOPT was also modified to yield a smoother convergence to the problem solution and this is discussed in subsection 4.1.3. The final modification discussed in subsection 4.1.4, provided the option of using a different variable metric update to yield a more accurate Hessian approximation

4.1.1. Implementation of a Factorized BFS Variable Metric Update

Variable metric updates have been successfully used for the past 20 years for unconstrained optimization and have been used successfully for approximately the past 10 years for constrained optimization. Variable metric updates attempt to build an approximation to the Hessian matrix using only first order information, and solve for the search direction from the following equation

$$\mathbf{s} = \mathbf{B}^{-1} \nabla f \quad (4.1)$$

where \mathbf{B} represents the approximation to the Hessian, ∇f the gradient of the objective function, and \mathbf{s} the search direction of the design variables. Variable metric updates have been provided in the literature for approximating either the inverse of the Hessian or the Hessian itself.

Variable metric updates all have the same basic form. They begin with an approximation to the Hessian matrix, and then update the approximation by some rank one or rank two correction. The form of the update is normally

$$\mathbf{B}_{\text{new}} = \mathbf{B}_{\text{old}} + \mathbf{v}\mathbf{v}^T + \mathbf{w}\mathbf{w}^T \quad (4.2)$$

where \mathbf{v} and \mathbf{w} are calculated as some product of the old Hessian approximation, the last search direction, and the change in the gradient of the objective function.

Several different forms of equation 4.2 have been proposed. The most popular

variable metric update has been the BFS (also known as the BFGS) which was proposed in 1970 simultaneously by Broyden, Fletcher, Shanno and Goldfarb. The BFS update has been shown to be the best general purpose variable metric update.

One of the problems associated with the BFS variable metric update is that it is effected by the problem scaling. Shanno and Phua (1978) have proposed a self scaling version of the BFS update. Its use in a RQP algorithm was investigated by Van der Hoek (1980). He found the self scaling variant with the second Oren-Spedicato (Oren 1974) switch seemed to perform the best with the particular RQP algorithm that he was using.

In the mid 1970's several authors proposed updating the LDL^T factors of the Hessian approximation with a procedure that could be used to stabilize the BFS update in terms of the numerical noise encountered in the calculation of the update. With the LDL^T update we can be assured that the Hessian approximation remains positive definite, this will assure that the search directions that are generated from (4.1) are downhill. Additionally, finding the search direction from equation 4.1 becomes a simple matrix calculation when using the LDL^T update

When variable metric updates are used for RQP methods it is normally preferred that the approximation of the Hessian of the Lagrangian be updated instead of its inverse. This is because solution of the QP subproblem requires the Hessian approximation. The BFS variable metric update is used by most of the successful implementations of the RQP method.

The BFS update that was used in RQOPT is defined as

$$\mathbf{B}_{\text{new}} = \mathbf{B}_{\text{old}} - \frac{(\mathbf{z}\mathbf{B}_{\text{old}})(\mathbf{z}\mathbf{B}_{\text{old}})^T}{\mathbf{z}^T\mathbf{B}_{\text{old}}\mathbf{z}} + \frac{\mathbf{w}\mathbf{w}^T}{\mathbf{w}^T\mathbf{y}} \quad (4.3)$$

where \mathbf{z} and \mathbf{w} are defined as

$$\mathbf{z} = \mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}} \quad (4.4)$$

$$\mathbf{y} = \nabla_{\mathbf{x}}L(\mathbf{x}_{\text{new}}, \mathbf{v}_{\text{new}}, \mathbf{u}_{\text{new}}) - \nabla_{\mathbf{x}}L(\mathbf{x}_{\text{old}}, \mathbf{v}_{\text{new}}, \mathbf{u}_{\text{new}}) \quad (4.5)$$

$$\Theta = \begin{cases} 1 & \text{if } \mathbf{z}^T\mathbf{y} \geq 0.2 \mathbf{z}^T\mathbf{B} \mathbf{z} \\ \frac{0.8 \mathbf{z}^T\mathbf{B} \mathbf{z}}{\mathbf{z}^T\mathbf{B} \mathbf{z} - \mathbf{z}^T\mathbf{y}} & \text{otherwise} \end{cases} \quad (4.6)$$

$$\mathbf{w} = \Theta \mathbf{y} + (1 - \Theta)\mathbf{B}\mathbf{z} \quad (4.7)$$

Where the Θ term in equation 4.6 and 4.7 was defined by Powell (1977) to help maintain positive definiteness of the Hessian approximation, under normal operation Θ is equal to one. The Hessian approximation is guaranteed to be positive definite if $\mathbf{z}^T\mathbf{w}$ is greater than

zero. The Hessian approximation is not updated by RQOPT if $\mathbf{z}^T \mathbf{w}$ is less than zero.

For this study, the LDL^T update for the BFS variable metric (defined in equation 4.3) as described by Gill and Murray (1978) was implemented (where \mathbf{z} and \mathbf{w} were calculated by equations 4.5 and 4.7). This update uses several matrix transformations to achieve a stable update. The actual update of the Hessian approximation is performed with a procedure described by Fletcher and Powell (1974) and extended by Gill, Murray, and Saunders (1975).

In addition to the stability of this update relative to numerical noise, as discussed above, the LDL^T update provides a convenient means for establishing a reset criteria for the Hessian approximation. The need for a reset of the Hessian approximation is discussed in the following section.

4.1.2 Condition Number Reset

Occasionally, due to numerical noise or a highly non linear problem, the Hessian approximation may become singular or indefinite. When this happens we can no longer be certain that the resulting search directions will satisfy the descent property that is assumed by the RQP. The only means to recover from this situation is to reset the approximation to some known positive definite matrix, which is generally the identity matrix. Early version of the BFS update were reset every $n+1$ iterations but this is a conservative approach that will sometimes erase good information and slow the convergence of the algorithm. The current thinking is to use a less conservative reset criteria that is based on a condition number estimate of the matrix with the hope that useful information built up in previous iterations is used for more iterations and should result in better convergence.

The original version of RQOPT reset the Hessian approximation every time the active set changed or every $n+1$ iteration. A change in the active set results in a different QP subproblem to be solved and it was felt that the Hessian approximation would no longer be valid. Using this conservative reset criteria would prove unacceptable if we were using RQOPT to perform sensitivity analysis. With this reset criteria, we risk resetting the Hessian approximation just before the optimum is reached and would be left with only a few iterations of the method upon which to build an approximation. Thus we may have a very poor Hessian approximation when it comes time to perform the sensitivity analysis.

The reset criteria adopted has been used successfully by several other algorithms (Powell 1985, Schittkowski 1983, Arora and Tseng 1987). The new reset criteria resets the Hessian approximation when the estimate of the condition number exceeds a fixed limit. This estimate can be found by computing

$$\text{cond}(\mathbf{H})_{\text{est}} = \frac{d_{\text{max}}}{d_{\text{min}}} \quad (4.8)$$

where d_{min} and d_{max} are the smallest and largest values of the \mathbf{D} matrix in the LDL^T factorization.

Using this reset criteria has led to a more stable update yielding faster convergence for the RQOPT program and more accurate estimates of the Hessian of the Lagrangian.

4.1.3 Calculation of the Lagrange Multiplier Estimates

The Lagrange multiplier estimates are an integral part of building the Hessian approximation. The value of the Lagrange multiplier estimates are used as inputs to the variable metric update to approximate the Hessian of the Lagrangian function.

The original version of RQOPT calculated the Lagrange multiplier estimates as the Lagrange multipliers of the constraints in the QP subproblem. This value of the Lagrange multiplier estimate is a valid estimate of the true multipliers when a step of $\alpha = 1$ is used in the line search (Gill and Murray 1979). When this occurs, the estimates should converge to the true Lagrange multipliers as the problem converges.

A problem can arise, however, in the first few iterations of RQOPT. At the beginning of a search it is possible that a Lagrange multiplier estimates produced by the QP subproblem will be several orders of magnitude larger than true value of the Lagrange multiplier. If the line search then makes a small step ($\alpha \ll 1$), the large value of the Lagrange multiplier estimate may bias the updating of the Hessian approximation in such a way that new approximation only sees the constraint associated with the large Lagrange multiplier. It may then take several iterations before the Hessian approximation is corrected.

RQOPT was modified to use the following linear interpolation to update the value of the Lagrange multiplier estimates after the line search is completed

$$\mathbf{u}_{\text{new}} = \mathbf{u}_{\text{old}} + \alpha(\mathbf{u}_{\text{qp}} - \mathbf{u}_{\text{old}}) \quad (4.9)$$

When a step length of $\alpha = 1$ is used in the line search (equation 3.4) then formula 4.9 updates the Lagrange multiplier estimates to be the estimates delivered by the QP subproblem. This update was also used by Schittkowski (1983).

The procedure for updating the Lagrange multiplier estimates helped yield a smoother convergence of the Hessian approximation, because we were able to more accurately represent the Lagrangian function when we were performing the approximation updates.

4.1.4. SR1 update

The SR1 update is a variable metric update that does not require exact line searches for quadratic convergence, where as the BFS update requires exact line searches for quadratic convergence. Because the RQP method seldom performs exact line searches, it was felt the SR1 update may be able to obtain a better approximation of the Hessian of the Lagrangian.

A table describing the differences between the BFS and SR1 update is presented below

	Advantages	Disadvantages
update BFS	Self Correcting Stable (maintains positive definiteness) Has a good performance history	Requires exact line searches
SR1	Does not require exact line searches	update may be undefined and it is not guaranteed to maintain positive definiteness of the Hessian Approximation. There is not alot of literature on the performance of this update.

Table 4.1 A comparison of the BFS and SR1 variable metric updates

The stability of the BFS variable metric update has led to its use in almost all RQP implementations. However Cha and Mayne (1987) report that they have tested the SR1 update and found exact convergence of Hessian approximations for quadratic functions. Although the SR1 update lacks the stability of the BFS update, we were interested in comparing the performance of the 2 updates in terms of the Hessian convergence. If the SR1 update delivers better Hessian approximations than the BFS update then we will have to further investigate methods to stabilize the SR1 update.

The SR1 update is defined as follows

$$\mathbf{B}_{\text{new}} = \mathbf{B}_{\text{old}} + \frac{(\mathbf{B}_{\text{old}}\mathbf{y} - \mathbf{z})(\mathbf{B}_{\text{old}}\mathbf{y} - \mathbf{z})^T}{\mathbf{y}^T(\mathbf{B}_{\text{old}}\mathbf{y} - \mathbf{z})} \quad (4.10)$$

where \mathbf{y} and \mathbf{z} are obtained from equation 4.4 and 4.5. This update is undefined when the denominator is equal to zero. The SR1 update may be undefined even for positive definite quadratic problems. This problem was addressed by Brayton and Cullem (1979), Cullem and Brayton (1979).

The symmetric rank one (SR1) update was implemented in both a factored (LDL^T) and unfactored form. In our implementation if the absolute value of the denominator (in equation 4.10) is less than some small number we use the BFS update which is described in section 4.1.1.

Even though the SR1 update may be undefined, it has the very nice property of not requiring exact line searches. This is important because in the RQP method we do not perform exact line searches, and the BFS variable metric method assumes exact line searches. Powell (1986) clearly demonstrates the detrimental effect of inexact line searches on the BFS method. The performance of the SR1 update for solving quadratic problems is such that after n updates (providing that all updates are defined) the Hessian approximation will have converged to the true Hessian. Thus we may obtain a better convergence of the approximation of the Hessian of the Lagrangian if we are able to use the SR1 update.

Some preliminary results were obtained comparing the BFS and SR1 variable metric updates and these are discussed in section 5.3.

4.2 THE CREATION OF A SYSTEM TO AUTOMATICALLY CALCULATE PARAMETER SENSITIVITIES

In this section we provide a brief overview of the software system created for studying parameter sensitivities. The software system is made up of three major pieces: a problem preparation package RQCRE, the RQP algorithm using the modifications described in the preceding section, RQOPT, and an interactive program RQSEN, that acts as a post processor/sensitivity analysis module for the RQOPT program. The RQOPT program was an existing program and has been documented previously (Beltracchi and Gabriele, 1986). The RQCRE and RQSEN programs were created for this study and will be briefly described in the following paragraphs. A more detailed discussion of these systems is provided in the appendix.

4.2.1 The RQCRE Support System

The RQCRE program is set up to be used as an interactive tool for use with the RQSEN system. The purpose of the RQCRE program is to remove the chance of errors in the problem formulation. The RQSEN program requires approximately 30 arrays to be dimensioned which are automatically dimensions by RQCRE. The RQCRE program also automatically writes the calling program and data files required by the RQSEN system.

The RQCRE program requires the user to provide basic information about the problem such as the number of variables, number of equality constraints, number of inequality constraint, and number of parameters that will be studied.

The RQCRE system then produces a main calling program, a shell of the function subprogram¹ used to define the objective function and the constraints, and a data file used for input into the RQSEN system (sample output is provided in the appendix). The

¹ The RQCRE program is not designed to allow the user to enter definitions of the objective function or constraints, these definitions must be entered manually into the code that was generated by RQCRE.

RQCRE program also sets up the default values for the algorithm parameters used by RQOPT.

4.2.2 The RQSEN program

The RQSEN system was set up as a pre and post processor for the RQOPT program. The RQSEN system was set up to be an interactive user friendly program for performing the following basic functions;

1. The system can be used to solve optimal design problems
2. The system can be used to calculate parameter sensitivities
3. The system can be used to conduct studies of large variations in problem parameters
4. The system is also set up to create sensitivity plots of that can be used to perform trade off studies.

A sample session with the RQSEN system illustrating these options is presented in the appendix.

The RQSEN system requires a calling program and a function subprogram (defining the objective function and the constraints) to be written in FORTRAN². The RQSEN system also requires the user to define a data file that contains the algorithm parameters, and the initial values of the design variables and design parameters. The user can then direct the RQSEN program to study the sensitivities of only certain parameters.

The RQSEN program first produces optimum designs. Once the problem has been optimized the RQSEN system can be used to produce parameter sensitivity derivatives, which can then be used to study the effect on the optimum of large variations in the parameters. The RQSEN system is also set up so that an external graphing program can be used to create plots of the optimum sensitivities for large variations in the parameters can be studied. A typical plot is presented in figure

² The RQCRE system can be used as an aid in creating the calling program and function subprogram.

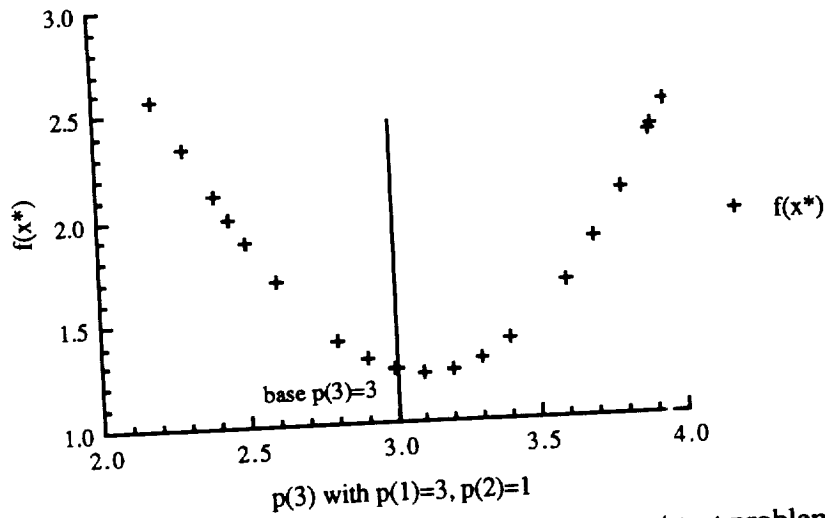


Figure 4.1 A plot of the Sensitivity of the Optimum of test problem 1 to $p(3)$

Plots similar to this one can also be generated for the design variables, Lagrange multipliers and values of the constraints. These plots can then be used to assess the characteristics of the problem (such as nonlinearity and changes in the active set). Using these plots to assess the characteristics of the problem will be discussed in the results part of chapter 5. Plots similar to figure 4.1 are presented in the appendix for problems in the test set.

5. Numerical Experiments

This chapter describes the numerical experiments that have been conducted to date on the new sensitivity method. We begin by discussing the initial test set used and any special features of the selected problems. Next, we discuss testing that has been performed comparing the accuracy of the known Hessian to the approximations obtained, which includes comparisons of the BFS and SR1 updates. In the third section, the accuracy of the sensitivity derivatives obtained with the new sensitivity algorithm is assessed against the known results. This section also compares the effect of choosing a central or forward difference formula and the effect of the step size Δp . The final section presents some conclusions drawn from this initial testing.

5.1. INITIAL TEST SET

A two phase testing program has been formulated for studying the effectiveness of the new method for estimating parameter sensitivity. The first phase was to develop a set of test problems for which the parameter sensitivities could be exactly determined using the Kuhn-Tucker equations. This required that any second order information needed could be determined analytically. Choosing problems of this type would allow a direct comparison of the sensitivity results produced by the new method with the exact sensitivities and also allow the comparison of the BFS and SR1 Hessian approximations. From this study we hope to develop some insight into several questions concerning the algorithm such as: proper choices for algorithm parameters (i.e. the proper size of Δp), what is the most reliable Hessian approximation, how close does the Hessian approximation have to be to achieve good results, does updating the Hessian approximation during the sensitivity analysis significantly improve the estimate, and which of the variants (forward/central difference approximations with one or two iterations of RQOPT) described in chapter 3 provides the most consistent results.

The second phase of the testing would consist of testing the algorithm against a set of engineering problems where second order information would not be available. Here the results obtained from the sensitivity algorithm would be compared to actual reoptimization results to assess its accuracy. In the time allotted for this study, only the first phase of testing has been completed and is reported on here.

The problems making up the initial test set are presented in the appendix of this report. We have experimented so far with 4 test problems that have a total of 12 parameters. The problems possess both linear and nonlinear behavior. We expect to expand this test set in the near future. Plots of the optimum sensitivity for selected

problems and parameters are also presented in the appendix.

5.2. CONVERGENCE OF THE HESSIAN APPROXIMATION

The derivation given in section 3.2.1 showed that equivalence of the new method with the Kuhn-Tucker method depends on the accuracy of the Hessian approximation obtained from the RQP method. Using this initial test set we hope to observe how closely the Hessian approximation comes to the exact Hessian and draw some initial conclusions on its importance to the accuracy of the results.

A measure of the closeness of the Hessian approximation to the true Hessian can be defined using the Frobenius norm as

$$\epsilon_H = \| H - H_{\text{approx}} \|_F \quad (5.1)$$

This measure has been used in the past to compare the convergence of different variable metric updates (Dennis and Schnable 1983).

For test problem 1 the true Hessian of the Lagrangian is

$$H = \begin{bmatrix} 2.64 & 0 \\ 0 & 2.6 \end{bmatrix}$$

From the RQOPT program we obtained the following Hessian approximation with the BFS update

$$H_{\text{BFS}} = \begin{bmatrix} 1.50017 & -.540310 \\ -.540310 & 2.34388 \end{bmatrix}$$

which gave us a $\epsilon_{H_{\text{BFS}}} = 1.396$.

Using the SR1 update from the same starting point, we obtained the following Hessian approximation

$$H_{\text{SR1}} = \begin{bmatrix} 2.63194 & -.002930 \\ -.002930 & 2.61374 \end{bmatrix}$$

with gives a $\epsilon_{H_{\text{SR1}}} = 0.0164$. This represents a large improvement in the closeness of the Hessian approximation. However, even though the Hessian approximation for the SR1 update is much better than the Hessian approximation for the BFS update the problems were solved in the same number of iterations (and function/constraint evaluations) of RQOPT.

The results given above were obtained with a value of $\delta = 1.1$. The δ parameter controls the size of the active set during the course of an optimization; a large δ will cause more near active constraints to be considered as part of the active set, a small value of δ will

allow only truly active constraints to be considered. Having the proper set of active constraints identified early in the optimization could effect the accuracy of the Hessian approximation. To test this, a larger value of δ ($\delta = 10.1$) was chosen and the problem resolved obtaining the following Hessian approximations

$$\mathbf{H}_{\text{BFS}} = \begin{bmatrix} 2.329 & .3227 \\ .3227 & 2.264 \end{bmatrix}$$

$$\mathbf{H}_{\text{SR1}} = \begin{bmatrix} 2.6394 & .00093 \\ .00093 & 2.5990 \end{bmatrix}$$

the values of $\epsilon_{\text{HBFS}} = 0.6019$ and $\epsilon_{\text{HSR1}} = 0.00174$ were obtained. These improved Hessian approximations result because RQOPT was able to identify the correct active set of constraints sooner. With the large value of δ RQOPT required the same number of iterations to solve the problem, but required more constraint evaluations.

Another implementation issue that needs investigation concerns whether the Hessian approximation obtained from the optimization should be further updated during the reoptimizations performed to estimate the sensitivities. To study the effect of allowing Hessian updates during the reoptimization, the sensitivity with respect to parameter 3 in problem 1 was estimated with this option enabled. The Hessian approximation that was used at the start of the sensitivity analysis is the Hessian approximation that was obtained with the BFS update and $\delta = 1.1$. After estimating the sensitivity, we obtained the following Hessian approximation

$$\mathbf{H}_{\text{BFS}} = \begin{bmatrix} 2.63975 & .00013 \\ .00013 & 2.59957 \end{bmatrix}$$

with $\epsilon_{\text{HBFS}} = 0.00053$. This indicates that there is a possibility for improving the Hessian approximation if we allow updating during the sensitivity analysis.

Tests for problem 2 were also performed, whose true Hessian of the Lagrangian is given by

$$\mathbf{H} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \end{bmatrix}$$

Using the starting point provided in the problem description, we obtain the following value of the Hessian approximation (from RQOPT) when we use the BFS update

$$\mathbf{H}_{\text{BFS}} = \begin{bmatrix} 4.4976 & 1.9976 & 0.49763 \\ 1.9976 & 2.9976 & 1.9976 \\ 0.49763 & 1.9976 & 4.4976 \end{bmatrix}$$

with $\epsilon_{\text{HBFS}} = 3.000$. When we use the SR1 update we obtain the following value of the Hessian approximation

$$\mathbf{H}_{\text{SR1}} = \begin{bmatrix} 4.5 & 2.0 & 0.5 \\ 2.0 & 3.0 & 2.0 \\ 0.5 & 2.0 & 4.5 \end{bmatrix}$$

$\epsilon_{\text{H}_{\text{SR1}}} = 3.0$. If we allow the Hessian matrix to be updated while estimating the sensitivity of p_1 with a $\Delta p = 0.0001$ we obtain

$$\mathbf{H}_{\text{BFS}} = \begin{bmatrix} 4.9448 & 1.0070 & 1.1749 \\ 1.0070 & 4.9964 & 0.9665 \\ 1.1749 & 0.9665 & 4.3990 \end{bmatrix}$$

$$\mathbf{H}_{\text{SR1}} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \end{bmatrix}$$

with $\epsilon_{\text{H}_{\text{BFS}}} = 0.6541$ and with $\epsilon_{\text{H}_{\text{SR1}}} = 0.00$. This represents a significant improvement of the Hessian approximations, particularly when the SR1 update is used.

If we calculate the sensitivity of p_2 and use the SR1 update we also obtain exact convergence of the Hessian approximation. However if we use the BFS update we do not obtain exact convergence but an improvement similar to that of the first problem is achieved.

For Test problem 3 the Hessian of the Lagrangian is

$$\mathbf{H} = \begin{bmatrix} 12 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

If we use the starting point that was provided in the problem description, the approximation to the Hessian of the Lagrangian (from RQOPT) using the BFS update is

$$\mathbf{H}_{\text{BFS}} = \begin{bmatrix} 9.785 & -0.4657 & -2.502 & -0.9879 \\ -0.4657 & 7.7556 & -0.5500 & 0.0174 \\ -2.502 & -0.5500 & 4.062 & 0.7464 \\ -0.9879 & 0.0174 & 0.7464 & 2.1579 \end{bmatrix}$$

with $\epsilon_{\text{H}_{\text{BFS}}} = 7.76$. If we use the SR1 update to solve the problem then we obtain the following approximation to the Hessian

$$\mathbf{H}_{\text{SR1}} = \begin{bmatrix} 11.9744 & -0.02493 & -0.04194 & 0.02212 \\ -0.02493 & 7.9792 & -0.03037 & 0.01095 \\ -0.04194 & -0.03037 & 9.9679 & 0.00222 \\ 0.02212 & 0.01095 & 0.00222 & 4.01399 \end{bmatrix}$$

with a $\epsilon_{\text{H}_{\text{SR1}}} = 0.130$. This represents a major improvement in the Frobenius norm.

For Test problem 4 the Hessian of the Lagrangian is

$$\mathbf{H} = \begin{bmatrix} 6.72 & -4.0 & -2.0 & 6.4 & -2.0 \\ -4.0 & 9.4006 & -1.2 & -6.2 & 6.4 \\ -2.0 & -1.2 & 4.4 & -1.2 & -2.0 \\ 6.4 & -6.2 & -1.2 & 9.3418 & -4.0 \\ -2.0 & 6.4 & -2.0 & -4.0 & 6.2688 \end{bmatrix}$$

using the starting point that was defined in the appendix, RQOPT with the BFS update yields the following Hessian approximation

$$H_{BFS} = \begin{bmatrix} 6.280 & -3.963 & -1.417 & 6.458 & -1.924 \\ -3.963 & 8.052 & -0.561 & -6.247 & 5.775 \\ -1.417 & -0.561 & 1.548 & -0.932 & -1.449 \\ 6.458 & -6.247 & -0.932 & 9.3465 & -4.024 \\ -1.924 & 5.775 & -1.449 & -4.024 & 5.974 \end{bmatrix}$$

with $\epsilon_{H_{BFS}} = 3.6226$.

When we attempted to use the SR1 update, the Hessian approximation became nearly singular after 5 iterations and the Hessian approximation was automatically reset to the identity matrix by RQOPT. RQOPT delivered the following Hessian approximation

$$H_{SR1} = \begin{bmatrix} 4.6611 & -4.8848 & 0.1290 & 5.2854 & -3.5329 \\ -4.8848 & 7.5178 & -0.1721 & -7.0517 & 4.7140 \\ 0.1290 & -0.1721 & 1.0046 & 0.1862 & -0.1245 \\ 5.2854 & -7.0517 & 0.1862 & 8.6328 & -5.0996 \\ -3.5329 & 4.7140 & -0.1245 & -5.0996 & 4.4094 \end{bmatrix}$$

with $\epsilon_{H_{SR1}} = 9.307$. The inaccuracy of this Hessian approximation results because a total of only 7 iterations were needed to solve the problem, and a reset occurred after the fifth iteration. Therefore, only 2 iterations could be used to build the Hessian approximation. In the near future we will investigate why the Hessian approximation became nearly singular after the 5th iteration.

A summary of the results of this section are presented in the Table 5.1 where ϵ_0 represents the error between the true Hessian and the identity matrix used at the outset of the optimization. Using the BFS update we see that we were not able to converge to the exact Hessian but the inaccuracies do not seem to be large. As mentioned before, this may be due to RQOPT not using exact line searches which the BFS method assumes. Allowing updating of the Hessian approximation during the sensitivity analysis seems to improve the estimate of the Hessian of the Lagrangian.

Using the SR1 update we were able to obtain better estimates of the Hessian of the Lagrangian for both problem 1 and 3. For problem 2 the Hessian of the Lagrangian that was produced by the SR1 update had converged in a projected or reduced sense. The inaccuracies in problem 4 are due to a near singular point which is discussed above.

Problem	ϵ_0	ϵ_{BFS}	ϵ_{SR1}	ϵ_{BFS} with updating
1	2.291	1.396	0.061	0.0005
2	7.348	3.0000	3.0	0.654
3	15.93	7.76	0.130	5.186
4	23.276	3.6226	9.307	1.698

Table 5.1 A comparison of the Frobenius norms of the Hessian approximations

5.3. RESULTS

This section presents a comparison of the sensitivity derivatives calculated by the new method with the known sensitivities for the problems in the initial test set. We also present a means that can be used to compare the accuracy of the sensitivity derivatives graphically.

The measure for accuracy that will be used was also used by Sandgren (1977). Sandgren compared the closeness of the optimum design point generated to known optimum point, and the closeness of the value of the known optimum objective function value to the generated value of the optimum plus a penalty for any violated constraints. Sandgren defined the following measures

$$\epsilon_f = \begin{cases} \text{ABS} \left[\frac{f(\mathbf{x}) - f(\mathbf{x}^*)}{f(\mathbf{x}^*)} \right] & \text{for } f(\mathbf{x}^*) \neq 0 \\ \text{ABS} [f(\mathbf{x})] & \text{for } f(\mathbf{x}^*) = 0 \end{cases} \quad (5.2)$$

where $f(\mathbf{x}^*)$ is the true value of the optimum and $f(\mathbf{x})$ is the value returned by the algorithm. The total error is calculated as

$$\epsilon_t = \epsilon_f + \sum_{j=1}^{n_{\text{eq}}} \langle g_j \rangle + \sum_{i=1}^{n_{\text{eq}}} (h_i) \quad (5.3)$$

where $\langle a \rangle = (0, \text{ if } a \geq 0 \mid -a \text{ if } a < 0)$. The ϵ_t measure is used because it does not bias any constraints.

The relative error in the \mathbf{x} vector is defined as

$$\epsilon_x = \sqrt{\sum_{i=1}^n \left[\frac{x_i - x_i^*}{x_i^*} \right]^2} \quad (5.4)$$

in equation 5.4 if x_i^* is equal to zero then the relative error in x_i is defined as the value of x_i .

We will define the relative error in the gradient (df^*/dp) of the objective function as follows

$$\epsilon_{df^*/dp} = \begin{cases} \text{ABS} \left[\frac{\frac{df^*}{dp}^{\text{est}} - \frac{df^*}{dp}}{\frac{df^*}{dp}} \right] & \text{for } \frac{df^*}{dp} \neq 0 \\ \text{ABS} \left[\frac{df^*}{dp}^{\text{est}} \right] & \text{for } \frac{df^*}{dp} = 0 \end{cases} \quad (5.5)$$

We will define the relative error in $\partial x^*/\partial p$ and $\partial u^*/\partial p$ in the same manner as ϵ_x and denote these values as $\epsilon_{\partial x^*/\partial p}$ and $\epsilon_{\partial u^*/\partial p}$ respectively. Eight digits of accuracy were maintained in calculating the relative errors.

The optimal sensitivities for the test problems were calculated using the Kuhn-Tucker method with exact derivatives. Once the optimal sensitivities for the problems were known, experiments were conducted using RQSEN on the initial test set. Both the forward difference and central difference variants of the RQP sensitivity algorithm were tested with large and small values of perturbation for the parameters. For all cases, RQOPT¹ was allowed to perform two iterations to optimize the perturbed problem. However, there were some instances where RQOPT required only one iteration to meet the convergence criteria. A spreadsheet was used to automate the calculation of the relative errors in the derivatives using the formulas given above. Summary tables showing the relative errors in the calculation of the derivatives will be presented for each of the problems.

Plots of the optimal sensitivities for large variations in the parameters were also created for all of the parameter sensitivities that were studied. The interesting plots will be included in the appendix of this report. These plots can be used to help assess the nonlinearity in the sensitivity derivatives, and to help to understand the effect of changes in the active set.

The rest of this section presents tables and figures showing the relative accuracy of the sensitivity derivatives. A brief discussion of the results for each problem is offered.

5.3.1 Problem 1

Problem 1 possesses three parameters for study. Sensitivities of the objective function, design variable, and Lagrange multipliers for each parameter were estimated using the four variations of the basic algorithm. The results are compared against the exact sensitivities in Tables 5.2 - 5.7. In most cases, the estimated sensitivities agree with the known sensitivities with few exceptions. As might be expected, the central difference approximations in all cases provides better estimates than the forward difference approximations. No strong conclusions with respect to the choice of Δp can be drawn from this problem. For parameter 1, both sizes of Δp provide exact sensitivities. For

¹The gradients of the objective function and constraints were calculated using central and forward difference approximations.

parameter 2, the larger value of Δp provides better results while for parameter 3, the smaller value of Δp provides the best results. A review of the sensitivity plot for this parameter (figure A.2) shows that the sensitivities for this parameter are nonlinear.

In conclusion, for this problem, using a central difference approximation with either step size for Δp resulted in no significant errors in the sensitivity estimates.

Kuhn Tucker		Central Difference Approximations			
	Method	$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	1.0000000	1.00000000	0.00E+00	1.00000000	0.00E+00
dx ₁ /dp	0.0000000	0.000000	0.000E+00	0.000000	0.000E+00
dx ₂ /dp	0.0000000	0.000000	0.000E+00	0.000000	0.000E+00
ϵ_x			0.00E+00		0.00E+00
du ₁ /dp	-0.2000000	-0.200000	0.000E+00	-0.200000	0.000E+00
du ₂ /dp	0.4000000	0.400000	0.000E+00	0.400000	0.000E+00
ϵ_u			0.00E+00		0.00E+00

Table 5.2 Central Difference Approximations to the Parameter Sensitivities for problem 1 parameter 1

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	1.00000000	1.060000000	-6.00E-02	1.003000000	-3.00E-03
dx ₁ /dp	0.00000000	0.0000000	0.000E+00	0.0000000	0.000E+00
dx ₂ /dp	0.00000000	0.0000000	0.000E+00	0.0000000	0.000E+00
ϵ_x			0.00E+00		0.00E+00
du ₁ /dp	-0.20000000	-0.2000301	-1.503E-04	-0.2006318	-3.159E-03
du ₂ /dp	0.40000000	0.39984547	3.863E-04	0.39693215	7.670E-03
ϵ_u			4.15E-04		8.29E-03

Table 5.3 Forward Difference Approximations to the Parameter Sensitivities for problem 1 parameter 1

Kuhn Tucker		<u>Central Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-0.3000000	-0.30000000	0.00E+00	-0.30000000	0.00E+00
dx ₁ /dp	0.1000000	0.10000011	-1.100E-06	0.10000359	-3.590E-05
dx ₂ /dp	-0.2000000	-0.20000030	3.000E-06	-0.20000320	-1.600E-05
ϵ_x			3.20E-06		3.93E-05
du ₁ /dp	-0.1304000	-0.13040090	-6.902E-06	-0.13040262	-2.009E-05
du ₂ /dp	0.0008000	0.00079973	3.363E-04	0.00080051	-6.325E-04
ϵ_u			3.36E-04		6.33E-04

Table 5.4 Central Difference Approximations to the Parameter Sensitivities for problem 1
parameter 2

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-0.3000000	-0.30000000	0.00E+00	-0.30000000	0.00E+00
dx ₁ /dp	0.1000000	0.10004811	-4.811E-04	0.10000681	-6.810E-05
dx ₂ /dp	-0.2000000	-0.20017630	-8.815E-04	-0.20000665	-3.325E-05
ϵ_x			1.00E-03		7.58E-05
du ₁ /dp	-0.1304000	-0.13061831	-1.674E-03	-0.12851547	1.445E-02
du ₂ /dp	0.0008000	0.00114555	-4.319E-01	0.00999551	-1.149E+01
ϵ_u			4.32E-01		1.15E+01

Table 5.5 Forward Difference Approximations to the Parameter Sensitivities for problem 1
parameter 2

Kuhn Tucker		<u>Central Difference Approximations</u>			
	Method	$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-0.4000000	-0.40000000	0.00E+00	-0.40000000	0.00E+00
dx ₁ /dp	1.2000000	1.19995460	3.783E-05	1.19999990	1.000E-06
dx ₂ /dp	0.6000000	0.60007142	-5.952E-05	0.60000018	-3.000E-07
ϵ_x			7.05E-05		1.04E-06
du ₁ /dp	0.4048000	0.40501856	-5.399E-04	0.40480055	-1.359E-06
du ₂ /dp	-0.5896000	-0.58972201	-2.069E-04	-0.58960030	-5.088E-07
ϵ_u			5.78E-04		1.45E-06

Table 5.6 Central Difference Approximations to the Parameter Sensitivities for problem 1 parameter 3

Kuhn Tucker		<u>Forward Difference Approximations</u>			
	Method	$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-0.4000000	-0.36400000	9.00E-02	0.39820000	9.00E-02
dx ₁ /dp	1.2000000	1.20029290	-2.441E-04	1.20001670	-1.392E-05
dx ₂ /dp	0.6000000	0.59484082	8.599E-03	0.59973857	4.357E-04
ϵ_x			8.60E-03		4.36E-04
du ₁ /dp	0.4048000	0.39514199	2.386E-02	0.40367242	2.786E-03
du ₂ /dp	-0.5896000	-0.57797893	1.971E-02	-0.59207284	-4.194E-03
ϵ_u			3.09E-02		5.03E-03

Table 5.7 Forward Difference Approximations to the Parameter Sensitivities for problem 1 parameter 3

5.3.2 Problem 2

Tables 5.8 - 5.11 present results obtained for the two parameters of problem 2. There is some significant disagreement between the known and estimated sensitivities for parameter 1 in each of the variations tested. The inaccuracies seem to occur due to the Hessian approximation not converging. The cause of this is likely due to RQOPT not using exact line searches, which the BFS variable metric update assumes. In this case, possibly the SR1 update would produce better results. The results for parameter 2 are excellent for all variations.

The major conclusion to draw from this problem is that the convergence of the Hessian approximation can be a critical factor in the success of the new method.

	Kuhn Tucker Method	<u>Central Difference Approximations</u>			
		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-7.0000000	-7.00000000	<i>-1.43E-11</i>	-7.00000000	<i>-1.43E-11</i>
dx ₁ /dp	2.0888889	2.1593080	-3.371E-02	2.1598250	-3.396E-02
dx ₂ /dp	-2.1666667	-2.1459360	9.568E-03	-2.1459879	9.544E-03
dx ₃ /dp	-0.9166667	-1.0143604	-1.066E-01	-1.0139152	-1.061E-01
ϵ_x			<i>1.12E-01</i>		<i>1.12E-01</i>
du ₁ /dp	-4.6666667	-4.5219200	3.102E-02	-4.5311680	2.904E-02
ϵ_u			<i>3.10E-02</i>		<i>2.90E-02</i>
dg ₂ /dp	-6.0000000	-6.1756450	-2.927E-02	-6.1738965	-2.898E-02
ϵ_g			<i>2.93E-02</i>		<i>2.90E-02</i>

Table 5.8 Central Difference Approximations to the Parameter Sensitivities for problem 2
parameter 1

	Kuhn Tucker Method	<u>Forward Difference Approximations</u>			
		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-7.0000000	-7.00000000	<i>-1.43E-11</i>	-7.00000000	<i>-1.43E-11</i>
dx ₁ /dp	2.0888889	2.2087420	-5.738E-02	2.2379818	-5.738E-02
dx ₂ /dp	-2.1666667	-2.1161414	2.332E-02	-2.1100841	2.332E-02
dx ₃ /dp	-0.9166667	-1.1367750	-2.401E-01	-1.1301356	-2.401E-01
ϵ_x			<i>2.48E-01</i>		<i>2.48E-01</i>
du ₁ /dp	-4.6666667	-4.4230730	5.220E-02	-4.3803787	5.220E-02
ϵ_u			<i>5.22E-02</i>		<i>5.22E-02</i>
dg ₂ /dp	-6.0000000	-6.43386710	-7.231E-02	-6.3725934	-7.231E-02
ϵ_g			<i>7.23E-02</i>		<i>7.23E-02</i>

Table 5.9 Forward Difference Approximations to the Parameter Sensitivities for problem 2
parameter 1

		Kuhn Tucker	<u>Central Difference Approximations</u>		
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	12.00000000	12.00000000	0.00E+00	12.00000000	0.00E+00
dx ₁ /dp	0.33333333	0.3333333	1.000E-08	0.3333333	1.000E-08
dx ₂ /dp	0.33333333	0.3333333	1.000E-08	0.3333333	1.000E-08
dx ₃ /dp	0.33333333	0.3333333	1.000E-08	0.3333333	1.000E-08
ϵ_x			1.73E-08		1.73E-08
du ₁ /dp	2.33333333	2.3333333	1.429E-08	2.3333335	-7.143E-08
ϵ_u			1.43E-08		7.14E-08
dg ₂ /dp	2.00000000	2.0000000	0.000E+00	2.0000000	0.000E+00
ϵ_g			0.00E+00		0.00E+00

Table 5.10 Central Difference Approximations to the Parameter Sensitivities for problem 2 parameter 2

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P=2. \%$	relative error	$\Delta P=0.1 \%$	relative error
df/dp	12.00000000	12.00000000	0.00E+00	12.00000000	0.00E+00
dx ₁ /dp	0.33333333	0.3333333	1.000E-08	0.3333333	1.000E-08
dx ₂ /dp	0.33333333	0.3333333	1.000E-08	0.3333336	1.000E-08
dx ₃ /dp	0.33333333	0.3333333	1.000E-08	0.3333333	1.000E-08
ϵ_x			1.73E-08		1.73E-08
du ₁ /dp	2.33333333	2.3333333	1.429E-08	2.3333335	1.429E-08
ϵ_u			1.43E-08		1.43E-08
dg ₂ /dp	2.00000000	2.0000000	0.000E+00	2.0000000	0.000E+00
ϵ_g			0.00E+00		0.00E+00

Table 5.11 Forward Difference Approximations to the Parameter Sensitivities for problem 2 parameter 2

5.3.3 Problem 3

Problem 3 possesses 3 parameters for study, and the results are presented in Tables 5.12 - 5.17. Again, the central difference approximations produce the better results, with the small step perturbation better than the large step perturbation for the first two parameters. The accuracy of the estimates are good in comparison with the known sensitivities.

Kuhn Tucker		<u>Central Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-8.0000000	-8.0000001	-8.75E-09	-8.0000001	-8.75E-09
dx ₁ /dp	-1.1471984	-1.1551824	-6.960E-03	-1.1485459	-1.175E-03
dx ₂ /dp	-0.3342830	-0.3360517	-5.291E-03	-0.3355475	-3.783E-03
dx ₃ /dp	-0.2279022	-0.2189132	3.944E-02	-0.2265911	5.753E-03
dx ₄ /dp	-3.5403608	-3.5312020	2.587E-03	-3.5390270	3.767E-04
ϵ_X			4.05E-02		6.99E-03
du ₁ /dp	-67.3428300	-67.5423370	-2.963E-03	-67.3399900	4.217E-05
du ₃ /dp	55.4605800	55.6363930	-3.170E-03	55.5155000	-9.903E-04
ϵ_U			4.34E-03		9.91E-04
dg ₂ /dp	-1.6600260	-1.6579776	1.234E-03	-1.6595000	3.169E-04
ϵ_g			1.23E-03		3.17E-04

Table 5.12 Central Difference Approximations to the Parameter Sensitivities for problem 3 parameter 1

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	-8.0000000	-8.0000001	-8.75E-09	-8.0000001	-8.75E-09
dx ₁ /dp	-1.1471984	-1.0434709	9.042E-02	-1.1427923	3.841E-03
dx ₂ /dp	-0.3342830	-0.3298696	1.320E-02	-0.3378248	-1.060E-02
dx ₃ /dp	-0.2279022	-0.2847166	-2.493E-01	-0.2292447	-5.891E-03
dx ₄ /dp	-3.5403608	-3.5084682	9.008E-03	-3.5376210	7.739E-04
ϵ_X			2.66E-01		1.27E-02
du ₁ /dp	-67.3428300	-62.2695100	7.534E-02	-66.8998810	6.578E-03
du ₃ /dp	55.4605800	52.1090250	6.043E-02	55.1460460	5.671E-03
ϵ_U			9.66E-02		8.68E-03
dg ₂ /dp	-1.6600260	-1.6647592	-2.851E-03	-1.6589239	6.639E-04
ϵ_g			2.85E-03		6.64E-04

Table 5.13 Forward Difference Approximations to the Parameter Sensitivities for problem 3 parameter 1

Kuhn Tucker		<u>Central Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.0000000	0.0000000	0.00E+00	0.0000000	0.00E+00
dx ₁ /dp	0.0000000	0.0000000	0.000E+00	0.0000000	0.000E+00
dx ₂ /dp	0.0000000	0.0000000	0.000E+00	0.0000000	0.000E+00
dx ₃ /dp	0.0000000	0.0000000	0.000E+00	0.0000000	0.000E+00
dx ₄ /dp	0.0000000	0.0000000	0.000E+00	0.0000000	0.000E+00
ϵ_x			0.00E+00		0.00E+00
du ₁ /dp	0.0000000	0.0000000	0.000E+00	0.0000000	0.000E+00
du ₃ /dp	0.0000000	0.0000000	0.000E+00	0.0000000	0.000E+00
ϵ_u			0.00E+00		0.00E+00
dg ₂ /dp	1.00000000	1.0000000	0.000E+00	1.0000000	0.000E+00
ϵ_g			0.00E+00		0.00E+00

Table 5.14 Central Difference Approximations to the Parameter Sensitivities for problem 3 parameter 2

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.0000000	-2.200E-13	2.200E-13	-1.660E-12	1.660E-12
dx ₁ /dp	0.0000000	4.150E-05	-4.150E-05	8.316E-04	-8.316E-04
dx ₂ /dp	0.0000000	-3.300E-05	3.300E-05	-6.601E-04	6.601E-04
dx ₃ /dp	0.0000000	-2.020E-05	2.020E-05	-4.050E-04	4.050E-04
dx ₄ /dp	0.0000000	-3.090E-05	3.090E-05	-6.180E-04	6.180E-04
ϵ_x			6.46E-05		1.29E-03
du ₁ /dp	0.0000000	8.840E-03	-8.840E-03	1.769E-01	-1.769E-01
du ₃ /dp	0.0000000	-1.270E-02	1.270E-02	-2.553E-01	2.553E-01
ϵ_u			1.55E-02		3.11E-01
dg ₂ /dp	1.00000000	1.0000100	-1.000E-05	1.0020100	2.010E-03
ϵ_g			1.00E-05		2.01E-03

Table 5.15 Forward Difference Approximations to the Parameter Sensitivities for problem 3 parameter 2

Kuhn Tucker Method	<u>Central Difference Approximations</u>			
	$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp -10.0000000	-9.99999986	1.4E-08	-9.99999986	1.4E-08
dx ₁ /dp 1.3057920	1.3055639	1.747E-04	1.3079005	-1.615E-03
dx ₂ /dp 0.5460588	0.5465549	-9.085E-04	0.5481210	-3.777E-03
dx ₃ /dp -1.0541310	-1.0523005	1.737E-03	-1.0520338	1.990E-03
dx ₄ /dp -2.3741690	-2.3704118	1.583E-03	-2.3720509	8.921E-04
ϵ_x		2.53E-03		4.65E-03
du ₁ /dp 55.4605800	55.4057600	9.884E-04	55.5658650	1.708E-03
du ₃ /dp -56.5052230	-56.4808140	4.320E-04	-56.5128710	-7.214E-03
ϵ_u		1.08E-03		7.41E-03
dg ₂ /dp 0.67758780	0.6766357	1.405E-03	0.6767562	1.227E-03
ϵ_g		1.41E-03		1.23E-03

Table 5.16 Central Difference Approximations to the Parameter Sensitivities for problem 3 parameter 3

Kuhn Tucker Method	<u>Forward Difference Approximations</u>			
	$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp -10.0000000	-9.99999986	1.4E-08	-9.99999986	1.4E-08
dx ₁ /dp 1.3057920	1.3726972	-5.124E-02	1.3151920	-7.199E-03
dx ₂ /dp 0.5460588	0.5453465	1.304E-03	0.5489030	-5.209E-03
dx ₃ /dp -1.0541310	-0.9886692	6.210E-02	-1.0457860	7.916E-03
dx ₄ /dp -2.3741690	-2.3442141	1.262E-02	-2.3672333	2.921E-03
ϵ_x		8.15E-02		1.23E-02
du ₁ /dp 55.4605800	56.8153210	-2.443E-02	55.5101970	-8.946E-04
du ₃ /dp -56.5052230	-56.9128710	-7.214E-03	-56.7627370	-4.557E-03
ϵ_u		2.55E-02		4.64E-03
dg ₂ /dp 0.67758780	0.6668760	1.581E-02	0.6757969	2.643E-03
ϵ_g		1.58E-02		2.64E-03

Table 5.17 Forward Difference Approximations to the Parameter Sensitivities for problem 3 parameter 3

5.3.4 Problem 4

The final problem in the test set contains 4 parameters, the estimated sensitivities are presented in Tables 5.18 - 5.23. As before, the central difference approximations produce the best results, and for this problem, the small step perturbation performs best. Excellent agreement was achieved for all the parameters in this problem

Kuhn Tucker		<u>Central Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.517404073	0.51740407	0.00E+00	0.51740407	0.00E+00
dx ₁ /dp	-0.40000000	-0.4000000	0.000E+00	-0.400000	0.000E+00
dx ₂ /dp	0.09729967	0.0972994	2.960E-06	0.097300	-3.710E-06
dx ₃ /dp	-0.20000000	-0.2000000	0.000E+00	-0.200000	0.000E+00
dx ₄ /dp	0.28602513	0.2860244	2.447E-06	0.286026	-3.077E-06
dx ₅ /dp	-0.06773997	-0.0677393	9.994E-06	-0.067741	-1.247E-05
ϵ_x			1.07E-05		1.34E-05
du ₃ /dp	0.40709971	0.4071024	-6.681E-06	0.407102	-6.731E-06
du ₅ /dp	-0.05662417	-0.0566248	-1.038E-05	-0.056625	-1.038E-05
du ₆ /dp	0.34730309	0.3473052	-6.191E-06	0.347305	-6.219E-06
du ₉ /dp	0.01908492	0.0190855	-3.280E-05	0.019086	-3.233E-05
ϵ_u			3.56E-05		3.52E-05

Table 5.18 Central Difference Approximations to the Parameter Sensitivities for problem 4 parameter 1

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.517404073	0.51740407	0.00E+00	0.51740407	0.00E+00
dx ₁ /dp	-0.40000000	-0.4000000	0.000E+00	-0.400000	0.000E+00
dx ₂ /dp	0.09729967	0.0973210	-2.193E-04	0.097289	1.053E-04
dx ₃ /dp	-0.20000000	-0.2000000	0.000E+00	-0.200000	0.000E+00
dx ₄ /dp	0.28602513	0.2860770	-1.814E-04	0.286000	8.713E-05
dx ₅ /dp	-0.06773997	-0.0677900	-7.388E-04	-0.067716	3.548E-04
ϵ_x			7.92E-04		3.80E-04
du ₃ /dp	0.40709971	0.4073857	-7.025E-04	0.407116	-4.073E-05
du ₅ /dp	-0.05662417	-0.0565732	8.997E-04	-0.056623	2.283E-05
du ₆ /dp	0.34730309	0.3475276	-6.464E-04	0.347315	-3.481E-05
du ₉ /dp	0.01908492	0.0190319	2.780E-03	0.019083	9.583E-05
ϵ_u			3.07E-03		1.12E-04

Table 5.19 Forward Difference Approximations to the Parameter Sensitivities for problem 4 parameter 1

Kuhn Tucker		<u>Central Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.306110869	0.30611087	0.00E+00	0.30611087	0.00E+00
dx ₁ /dp	0.00000000	0.00000000	0.000E+00	0.00000000	0.000E+00
dx ₂ /dp	-0.14711868	-0.1471186	6.797E-07	-0.147119	4.758E-07
dx ₃ /dp	0.00000000	0.00000000	0.000E+00	0.00000000	0.000E+00
dx ₄ /dp	-0.04916518	-0.0491649	4.841E-06	-0.049165	3.620E-06
dx ₅ /dp	0.09817962	0.0981794	2.332E-06	0.098179	1.742E-06
ϵ_x			5.42E-06		4.05E-06
du ₃ /dp	-0.05662417	-0.0566248	-1.063E-05	-0.056625	-1.038E-05
du ₅ /dp	0.05051545	0.0505156	-3.662E-06	0.050516	-5.424E-06
du ₆ /dp	-0.06156394	-0.0615644	-6.903E-06	-0.061564	-7.115E-06
du ₉ /dp	0.00240162	0.0024015	3.935E-05	0.002402	-2.373E-06
ϵ_u			4.15E-05		1.39E-05

Table 5.20 Central Difference Approximations to the Parameter Sensitivities for problem 4 parameter 2

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.306110869	0.30611087	0.00E+00	0.30611087	0.00E+00
dx ₁ /dp	0.00000000	0.00000000	0.000E+00	0.00000000	0.000E+00
dx ₂ /dp	-0.14711868	-0.1470723	3.150E-04	-0.147117	1.196E-05
dx ₃ /dp	0.00000000	0.00000000	0.000E+00	0.00000000	0.000E+00
dx ₄ /dp	-0.04916518	-0.0490525	2.292E-03	-0.049161	8.675E-05
dx ₅ /dp	0.09817962	0.0980709	1.107E-03	0.098176	4.190E-05
ϵ_x			2.56E-03		9.71E-05
du ₃ /dp	-0.05662417	-0.0570971	-8.352E-03	-0.056648	-4.279E-04
du ₅ /dp	0.05051545	0.0510986	-1.154E-02	0.050545	-5.817E-04
du ₆ /dp	-0.06156394	-0.0620483	-7.868E-03	-0.061589	-4.015E-04
du ₉ /dp	0.00240162	0.0024686	-2.790E-02	0.002405	-1.404E-03
ϵ_u			3.23E-02		1.63E-03

Table 5.21 Forward Difference Approximations to the Parameter Sensitivities for problem 4 parameter 2

		<u>Central Difference Approximations</u>			
Kuhn Tucker Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	1.183954566	1.18395457	0.00E+00	1.18395457	0.00E+00
dx ₁ /dp	-0.20000000	-0.2000000	0.000E+00	-0.200000	0.000E+00
dx ₂ /dp	0.09408231	0.0940821	2.689E-06	0.094082	2.317E-06
dx ₃ /dp	-0.35000000	-0.3500000	0.000E+00	-0.350000	0.000E+00
dx ₄ /dp	0.22881747	0.2288169	2.666E-06	0.228817	2.273E-06
dx ₅ /dp	0.02931310	0.0293137	-2.030E-05	0.029314	-1.723E-05
ϵ_x			2.06E-05		1.75E-05
du ₃ /dp	0.34730308	0.3473054	-6.709E-06	0.347305	-6.277E-06
du ₅ /dp	-0.06156394	-0.0615644	-8.089E-06	-0.061564	-7.115E-06
du ₅ /dp	0.72069515	0.7206968	-2.248E-06	0.720697	-2.456E-06
du ₅ /dp	0.15964688	0.1596474	-3.382E-06	0.159647	-3.570E-06
ϵ_u			1.13E-05		1.04E-05

Table 5.22 Central Difference Approximations to the Parameter Sensitivities for problem 4 parameter 3

		<u>Forward Difference Approximations</u>			
Kuhn Tucker Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	1.183954566	1.183954566	0.00E+00	1.18395457	0.00E+00
dx ₁ /dp	-0.20000000	-0.2000000	0.000E+00	-0.200000	0.000E+00
dx ₂ /dp	0.09408231	0.0941409	-6.226E-04	0.094083	-2.073E-06
dx ₃ /dp	-0.35000000	-0.3500000	0.000E+00	0.350000	0.000E+00
dx ₄ /dp	0.22881747	0.2289599	-6.226E-04	0.228818	-2.098E-06
dx ₅ /dp	0.02931310	0.0291757	4.687E-03	0.029313	1.559E-05
ϵ_x			4.77E-03		1.59E-05
du ₃ /dp	0.34730308	0.3485794	-3.675E-03	0.347369	-1.895E-04
du ₅ /dp	-0.06156394	-0.0614102	2.497E-03	-0.061557	1.153E-04
du ₅ /dp	0.72069515	0.7229895	-3.184E-03	0.720811	-1.611E-04
du ₅ /dp	0.15964688	0.1595192	8.000E-04	0.159641	3.620E-05
ϵ_u			5.52E-03		2.77E-04

Table 5.23 Forward Difference Approximations to the Parameter Sensitivities for problem 4 parameter 3

Kuhn Tucker		<u>Central Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.010389619	0.01038962	0.00E+00	0.01038962	0.00E+00
dx ₁ /dp	0.00000000	0.00000000	0.000E+00	0.000000	0.000E+00
dx ₂ /dp	-0.03975934	-0.0397594	-8.803E-07	-0.039759	-1.082E-06
dx ₃ /dp	0.00000000	0.00000000	0.000E+00	0.000000	0.000E+00
dx ₄ /dp	0.07614087	0.0761408	1.116E-06	0.076141	1.366E-06
dx ₅ /dp	0.15499104	0.1549911	-5.162E-07	0.154991	-6.452E-07
ϵ_x			1.51E-06		1.86E-06
du ₃ /dp	0.01908492	0.0190855	-2.950E-05	0.019086	-3.238E-05
du ₅ /dp	0.00240162	0.0024016	-7.745E-06	0.002402	-2.498E-06
du ₆ /dp	0.15964687	0.1596474	-3.382E-06	0.159647	-3.633E-06
du ₉ /dp	0.08386033	0.0838607	-4.031E-06	0.083861	-4.042E-06
ϵ_u			3.09E-05		3.29E-05

Table 5.24 Central Difference Approximations to the Parameter Sensitivities for problem 4 parameter 4

Kuhn Tucker		<u>Forward Difference Approximations</u>			
Method		$\Delta P = 2\%$	relative error	$\Delta P = 0.1\%$	relative error
df/dp	0.010389619	0.01038962	0.00E+00	0.01038962	0.00E+00
dx ₁ /dp	0.00000000	0.00000000	0.000E+00	0.000000	0.000E+00
dx ₂ /dp	-0.03975934	-0.0397539	1.374E-04	-0.039759	1.856E-05
dx ₃ /dp	0.00000000	0.00000000	0.000E+00	0.000000	0.000E+00
dx ₄ /dp	0.07614087	0.0761542	-1.744E-04	0.076143	-2.359E-05
dx ₅ /dp	0.15499104	0.1549782	8.265E-05	0.154989	1.123E-05
ϵ_x			2.37E-04		3.20E-05
du ₃ /dp	0.01908492	0.0193233	-1.249E-02	0.019097	-6.561E-04
du ₅ /dp	0.00240162	0.0024360	-1.432E-02	0.002403	-7.328E-04
du ₆ /dp	0.15964687	0.1599642	-1.988E-03	0.159663	-1.032E-04
du ₉ /dp	0.08386033	0.0841484	-3.435E-03	0.083875	-1.755E-04
ϵ_u			1.94E-02		1.00E-03

Table 5.25 Forward Difference Approximations to the Parameter Sensitivities for problem 4 parameter 4

5.3.5 A Graphical Comparison of the Accuracy Delivered by Several Different Methods

This section presents a graphical method that can be used to assess the accuracy of the parameter sensitivity derivatives calculated by various methods. When we plot a bar chart of the calculated sensitivity derivatives versus the true sensitivity derivatives, we can identify which components of the gradient are least accurate. A brief discussion of the

significance of each graph will be provided.

Figure 5.1 and 5.2 present a graphical comparison of the $\partial x/\partial p_1$ and $\partial u/\partial p_1$ (as calculated by various methods²) for problem 3. In addition to results discussed in section 5.3.3, we calculated sensitivities by the modified Kuhn-Tucker method. The Hessian approximations are provided in section 5.2.

We can see in figure 5.1 that for large values of Δp for forward differencing we did not obtain results as accurate as those delivered by the other variants of the RQP sensitivity algorithm. If we examine the bars for the KT w/ BFS Hess we see that there is some discrepancy in the calculated values of the sensitivity derivatives. We also can see that using the Hessian approximation delivered by using the SR1 update we were able to obtain results better than those obtained by using the BFS approximation.

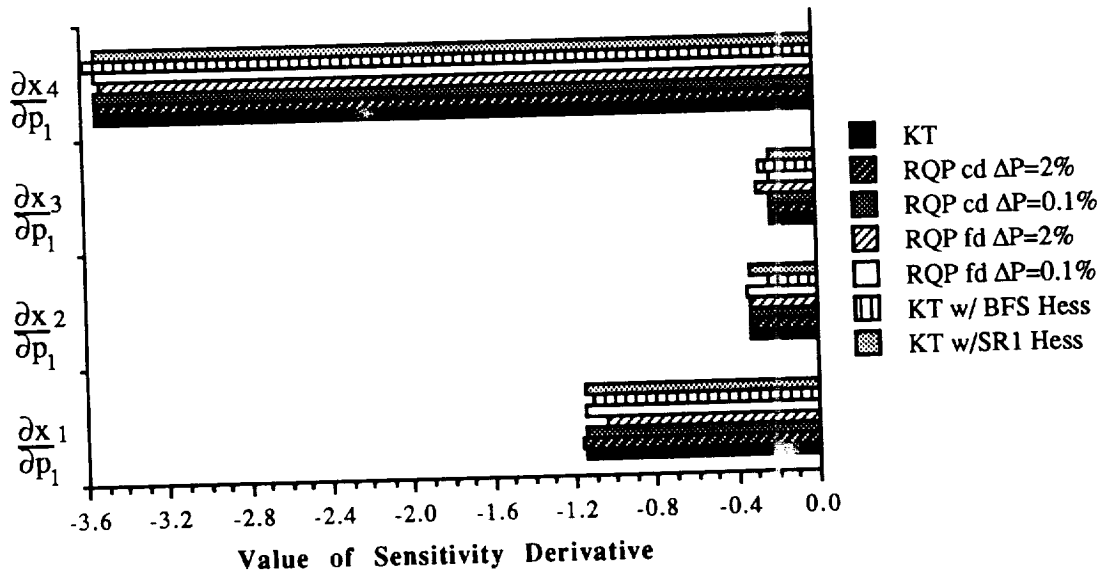


Figure 5.1 A comparison of the accuracy of $\partial x/\partial p$ for Problem 3 parameter 1

We can see in figure 5.2 that all but the KT w/ BFS Hess method produces good estimates of the sensitivity derivatives. The discrepancy in $\partial u/\partial p$ is due to the Hessian

²KT - Kuhn Tucker sensitivity equations with exact derivatives

RQP cd $\Delta P=2\%$ - RQP algorithm using 2 iterations to solve perturbed problem with central difference approximations $\Delta p=2\%$ of the nominal value

RQP cd $\Delta P=0.1\%$ - RQP algorithm using 2 iterations to solve perturbed problem with central difference approximations $\Delta p=0.1\%$ of the nominal value

RQP fd $\Delta P=2\%$ - RQP algorithm using 2 iterations to solve perturbed problem with forward difference approximations $\Delta p=2\%$ of the nominal value

RQP fd $\Delta P=0.1\%$ - RQP algorithm using 2 iterations to solve perturbed problem with forward difference approximations $\Delta p=0.1\%$ of the nominal value

KT w/ BFS Hess - The Kuhn Tucker sensitivity equations were solved using the approximate Hessian delivered by RQOPT when using the BFS update

KT w/ SR1 Hess - The Kuhn Tucker sensitivity equations were solved using the approximate Hessian delivered by RQOPT when using the SR1 update

approximation not converging.

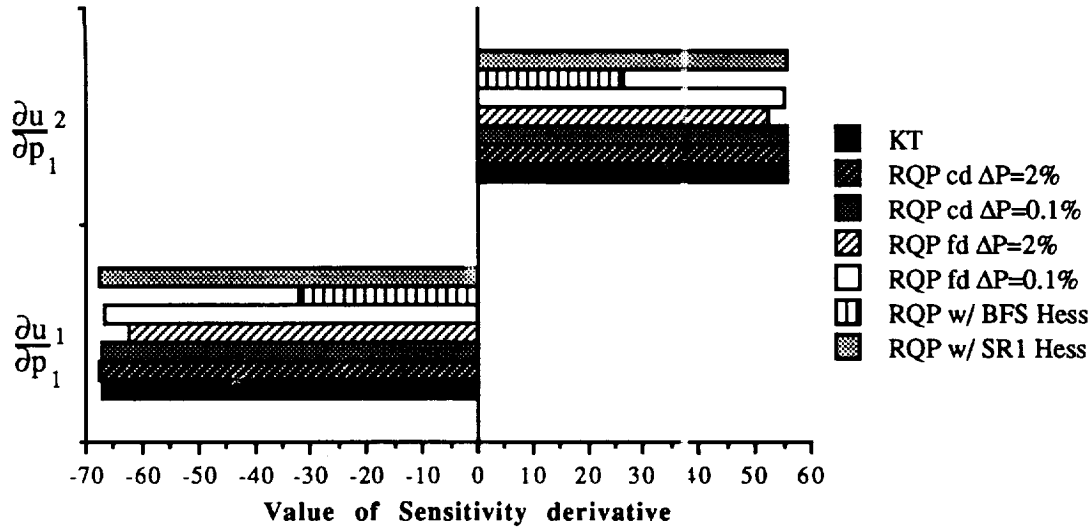


Figure 5.2 A comparison of the accuracy of $\partial u/\partial p$ for problem 3 parameter 1

Figures 5.3 and 5.4 present comparisons of the accuracy of the sensitivity derivatives (calculated by various methods²). We can see that all methods are in good agreement with the exception of the calculation of the KT with the approximate Hessian. Again the Hessian approximation did not converge fully for this problem.

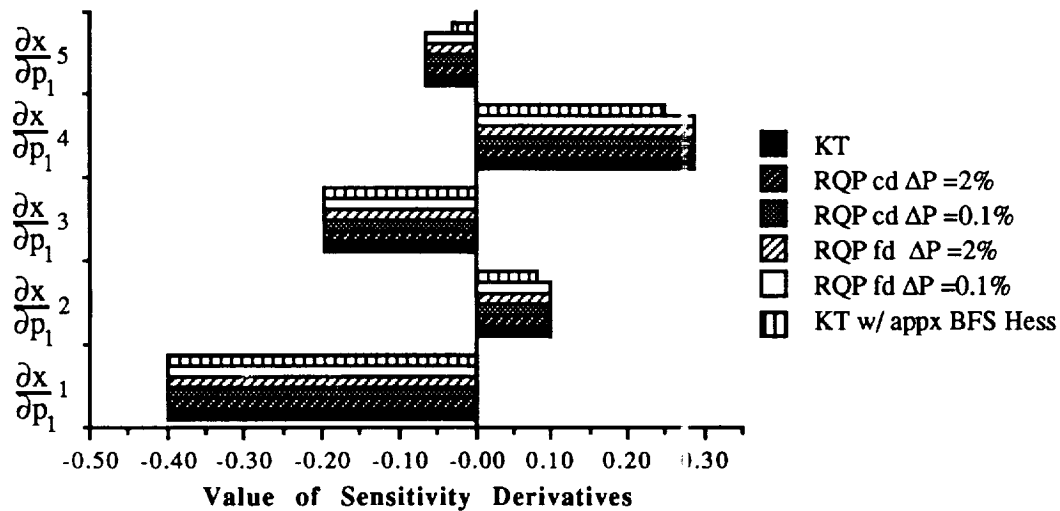


Figure 5.3 A comparison of $\partial x/\partial p$ for Problem 4 as calculated by various methods

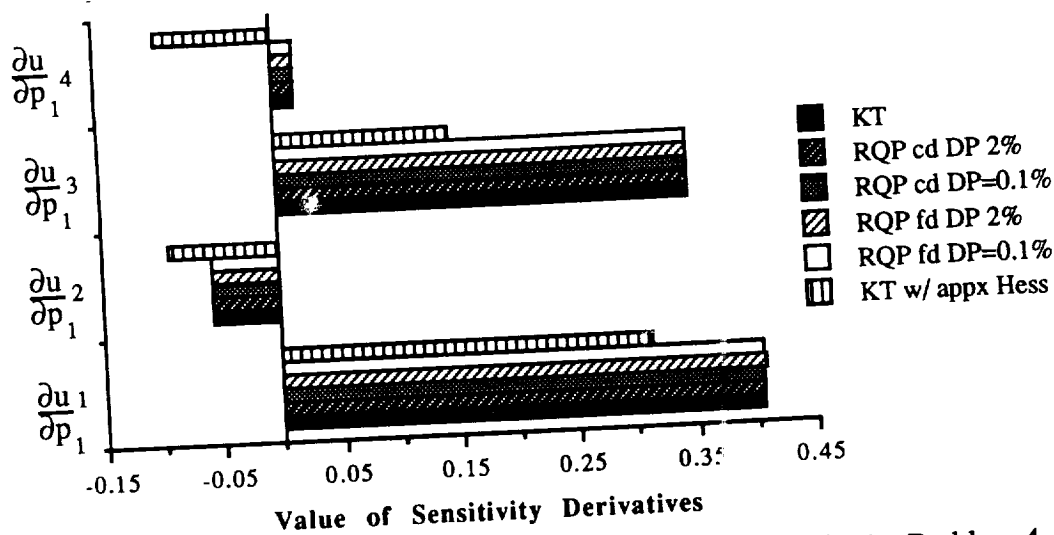


Figure 5.4 A comparison of various methods for calculation of $\partial u/\partial p$ for Problem 4

5.4. CONCLUSIONS FROM TEST RESULTS

In summary, we feel that this initial testing has shown that the RQP based method can produce reliable estimates of parameter sensitivities. Further testing is required to determine its performance on engineering problems, and to further resolve questions about algorithm parameters and variations. The following conclusions can also be drawn from this initial testing.

We saw in section 5.2 that the Hessian approximation improved if we allowed the approximation to be updated during the sensitivity analysis. During testing of the other problems in the test set we observed the Hessian approximation improving as we calculated the parameter sensitivities. This implies that if the Hessian approximation did not converge during the solution of the original problem (or converged in a projected sense) then a good estimate of the Hessian approximation can be built during the sensitivity analysis. Once we have a good approximation of the Hessian approximation then we can switch from a more expensive central differencing approximation to a less expensive forward difference approximation to the sensitivity derivative approximation. This conclusion is also encouraging because the RQP sensitivity algorithm approaches the Kuhn-Tucker sensitivity algorithm as the approximation improve (as was demonstrated in section 3.2).

If the Hessian approximation found by the RQP method is a good approximation of the Hessian of the Lagrangian then the Kuhn-Tucker sensitivity equations can be used with the approximate Hessian. However, tests need to be developed to check if the Hessian approximation has converged. If the Hessian approximation did not converge then updating it while using the RQP sensitivity algorithm can cause the Hessian approximation to converge. When this happens the Kuhn-Tucker sensitivity equations may be used with the Hessian approximation. Again, the issue to be resolved before this option can be

investigated is how to test a Hessian approximation for convergence.

Our experiments with the SR1 update were very encouraging. Section 5.2 demonstrated how the SR1 update was able to deliver a more accurate estimate of the Hessian approximation than the BFS update. We also observed exact convergence of the Hessian approximation with the SR1 update if we allowed the Hessian approximation to be updated during the sensitivity analysis. Near convergence of the Hessian approximation for problem 1 and 3 was also obtained when the SR1 update was used.

Even though our experiments with the SR1 update were encouraging the SR1 update has some serious drawbacks that will have to be investigated further before the method can be recommended. Unlike the BFS update the SR1 update is not self-correcting (Ip 1987). As we saw in problem 4, the SR1 update delivered an Hessian approximation that was singular, and the RQP method performs best if the Hessian approximation is positive definite.

In section 5.3, we observed that the method was able to approximate sensitivity derivatives. We saw that using central differencing yielded more accurate estimates of the sensitivity derivatives than using forward differencing approximations. We did not observe any major sensitivity to Δp when we were using the central differencing option.

We also observed that the forward differencing approximation has more trouble evaluating sensitivity derivatives, and the perturbation step size Δp had more effect when the functions being approximated are nonlinear in the parameter.

Finally, we observed that when the sensitivity derivative is small, the RQP sensitivity algorithm can sometimes have a difficult time finding the exact value. This was demonstrated in the calculation of $\partial u_2 / \partial p$ for problem 1 and also in the calculation of the sensitivity derivatives for p_2 in problem 3.

6. Changes in the Active Set

This chapter describes our findings regarding changes in the active set of constraints. The first section describes the four cases of changes in the active set of constraints. The second section provides sample plots illustrating the predicted behavior from section 1. Next, the third section discusses several proposed solutions for dealing with changes in the active set. The final section provides some procedures that can be used to generate problems where there are changes in the active set.

6.1. CASES OF ACTIVE SET CHANGE

When the active set of constraints change some of the sensitivity derivatives may be discontinuous. It is also possible that at the point where the active set changes, the gradients of the constraints may become linearly dependent. When this happens the optimization problem may become very difficult to solve. The four cases for changes in the active set are;

1. A constraint enters the active set (constraint gradients are linearly independent).
2. A constraint leaves the active set.
3. A constraint enters the active set (constraint gradients are linearly dependent).
4. A constraint enters the active set and causes there to be no feasible solution

Case 1 and Case 2 are complementary¹. This can best be demonstrated by an example. Consider an optimization problem whose solution changes as a parameter p , is varied and the active set changes as p increases. Assume the original problem is optimized for a $p=p^0$ and a sensitivity analysis is performed. The value of p is then increased to $p=p^1$ (where $p^1 > p^0$) and the problem is reoptimized. At $p=p^1$, constraint g_3 will enter the active set if p is increased any further, but the gradients of the constraints remain linearly independent as p is increased. Now p is increased to $p=p^2$ (where $p^2 > p^1 > p^0$) and at this point constraint g_3 has been added to the active set. A Case 1 active set change has occurred. To see the complementarity, assume we begin with $p=p^2$ and conduct a sensitivity analysis for decreasing values of p . Now when $p = p^1$, for any value of p less than p^1 constraint g_3 will leave the active set. Thus, when p reaches p^0 , constraint g_3 will have left the active set and we have had a Case 2 change in the active set for p going from p^2 to p^0 .

The behavior of the sensitivity derivatives for Cases 1 and 2 is characterized as

¹ This means that active set change algorithm that are developed for case 1 changes in the active set may also be used for case 2 changes in the active set

follows. There is a discontinuity in the rate of change of the objective function with respect to p (d^2f^*/dp^2 is discontinuous), there can be a discontinuity in the rate of change of some of the design variables ($\partial x^*/\partial p$ is discontinuous), and the rate of change of the Lagrange multipliers can be discontinuous (i.e. $\partial u^*/\partial p$ for a constraint leaving the active set will go from some nonzero value to zero, this may cause a change in the rate of change of the other Lagrange multipliers as well). The Hessian of the Lagrangian is continuous with respect to variations in p . For these problems there will be directional derivatives of $\partial x^*/\partial p$ and $\partial u^*/\partial p$ that can be used to make estimates of how the design variables and Lagrange multipliers will change. Second order extrapolations of the behavior of the objective function can also be made using d^2f^*/dp^2 in a directional sense.

The characteristic of Case 3 is a discontinuity in the Lagrange multiplier estimate (u^* is discontinuous). The discontinuity in the Lagrange multiplier causes a discontinuity in df^*/dp and also causes the Hessian of the Lagrangian to be discontinuous. Since the active set changes, there will also be a discontinuity in $(\partial x^*/\partial p)$. At the point where the constraints become linearly dependent it will not be possible to solve the Kuhn-Tucker sensitivity equations because they will become singular. For Case 3, there may be an exchange of constraints in the active set (i.e. the new constraint may replace one of the constraints that is already in the active set as p moves through the point).

The main characteristic of Case 4 is that there only exists a directional derivative away from the point where the path terminates. In order to calculate the directional derivative for Case 4 we need to be able to find the proper active set to follow when we leave the degenerate point. Case 4 changes in the active set will be common when the user overconstrains the design, i.e. sets the performance specifications to high to be physically meet by the given design configuration.

6.2. DEMONSTRATION OF CASES

This section describes the behavior that we observed in the initial test set for problems that had changes in the active set. This section also discusses two test problems that are not in the initial test set that are used to demonstrate the behavior of Case 3 changes in the active set.

The problems in the initial test set only had Case 1 and Case 2 changes in the active set. Indications of where the active set changes were shown on the plots of the optimum sensitivity provided in the appendix.

If we examine the plots that are presented in the appendix (figures A.1-A.8, A.10,

and A.11) we can see the discontinuity in $\partial x^*/\partial p$ and $\partial u^*/\partial p$ when the active set changes. In problems 3 and 4, because there is only a small discontinuity in $\partial x/\partial p$, it is difficult to see in the graphs provided (figures A.5,6,7,10,11) in the appendix. However for problems 3 and 4 it is easy to see the discontinuity in $\partial u/\partial p$, the slope of the Lagrange multipliers, when the active set of constraints changes.

If we examine figures A.4 and A.11, we see that not all of the components of $\partial x^*/\partial p$ are discontinuous when the active set changes. For problem 2 in figure A.4 we do not see a discontinuity in $\partial x_2^*/\partial p$, and for problem 4 in figure A.10 we do not see a discontinuity in $\partial x_1^*/\partial p$ or $\partial x_3^*/\partial p$, when the active set changes. This behavior is partially due to the fact that we were studying right hand side perturbations for these problems. The symmetry in problem 2 can be used to explain its behavior. In problem 4, the gradient of constraint g_9 had zero's in the first and third locations, thus we expect that perturbations of constraint g_9 should have no effect on $\partial x_1^*/\partial p$ or $\partial x_3^*/\partial p$.

The discontinuity in d^2f^*/dp^2 is difficult to see, when the active set changes for most of the problems in the initial test set. However for test problem 2, parameter 1, (figure A.3) we can see the discontinuity in d^2f^*/dp^2 . In the plot of the optimum value of the objective function versus p_1 we can see that for values of $p_1 \leq 1.055$, d^2f^*/dp^2 is less than zero (a region of negative curvature) and for values of $p_1 \geq 1.055$, d^2f^*/dp^2 is greater than zero (a region of positive curvature).

In problem 2 (p_2), problem 3 (p_1, p_2, p_3) and problem 4 (p_4) we studied the effect of perturbing the right hand sides of the inequality constraints. In these problems we studied a range of perturbations from where the constraint was inactive to where the constraint is in the active set. When the constraint was inactive perturbations in the parameters had no effect on the optimum. When the constraints entered the active set for these problems they caused d^2f^*/dp^2 to go from zero to some nonzero number. The discontinuity of d^2f^*/dp^2 can be seen in figures A.4, A.5, A.6, A.7, and A.11.

The following example will demonstrate how the optimum behaves for Case 3, when the gradients of the constraints become linearly dependent upon a change in the active set of constraints.

$$\text{minimize: } f = x_1^2 + (P - 1)^2 \quad (6.1)$$

$$\text{subject to: } g_1 = 3x_1 + 2P - 10 \geq 0 \quad (6.2)$$

$$g_2 = 2x_1 + 3P - 10 \geq 0 \quad (6.3)$$

when $P = 2$, the minimum $f^* = 5$ occurs at $x_1^* = 2$ with both constraints active and the gradients of the constraints linearly dependent. The Lagrange multipliers and in the family

$$u_1, u_2 \in \{3u_1 + 2u_2 = 4, u_1 > 0, u_2 > 0\} \quad (6.4)$$

For this problem df^*/dp , $\partial x^*/\partial p$ and $\partial u/\partial p$ do not exist. These derivatives do exist in both the positive and negative directions and are indicated by $\partial x^*/\partial p^+$ for increasing values of p and $\partial x^*/\partial p^-$ for decreasing values of p .

Figure 6.1 presents the sensitivity information for problem 6.1-6.3. Figure 6.1 (a) and (b) represents the first order predictions of the new values of the Lagrange multipliers for this problem. For this problem the linear predictions agree with the optimum Lagrange multipliers. There is a discontinuity at $\Delta p = 0.0$, therefore there are only be directional derivatives for these values. Figure 6.1 (c) represents linear predictions of the new value of the objective function. Notice again that there is a discontinuity in the slope of the prediction. Thus df^*/dp does not exist for this value of p (however df^*/dp^+ and df^*/dp^- will exist in a directional sense where the superscript + or - indicate the direction of change in p). Figure 6.1 (d) represents the predicted location of the optimum. Notice the discontinuity in the slope at $\Delta p = 0.0$, the true location of the optimum agrees with the linear prediction for this problem.

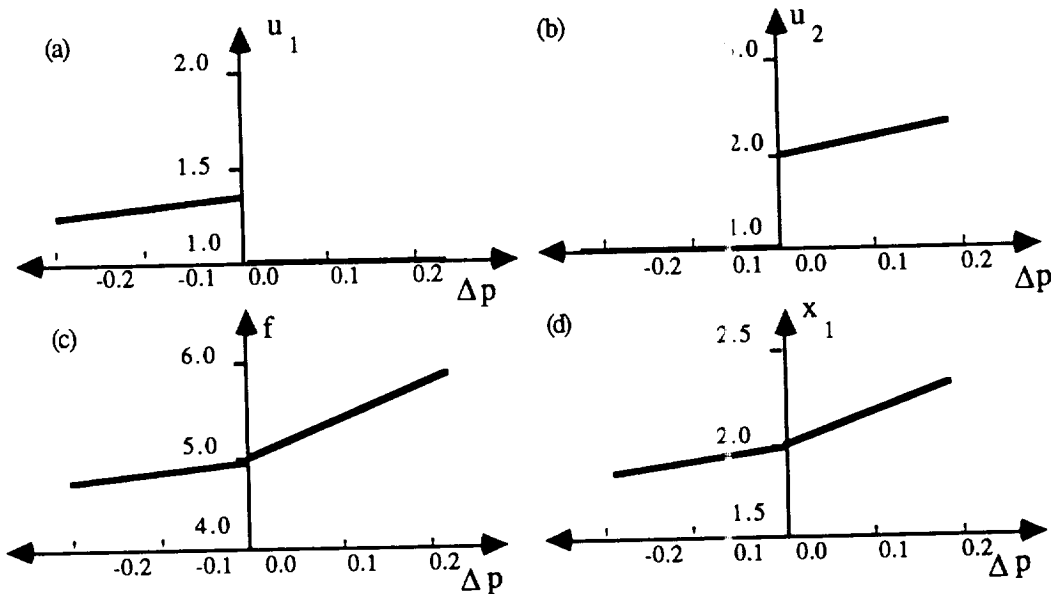


Figure 6.1 Plot of the optimum sensitivity predictions for problem 6.27-29

We have also examined a test problem used by Fiacco and Ghaemi (1982). This test problem has a linear dependence in the constraints gradients. The problem is to design a corrugated bulkhead for an oil tanker. The objective function is to minimize the weight of the bulkhead with constraints placed on allowable bending stress, moment of inertia, corrosion, and minimum gage thickness of the material. The problem has six design

variables, 18 design parameters, 17 constraints, a nonlinear objective function, and near linear constraints. We did not apply any scaling to this problem, even though the constraints for this problem are poorly scaled.

A sensitivity analysis was performed for variations in the parameter, LT , from 0.01% to 1% using the RQP code RQOPT (Beltracchi and Gabriele 1987). One step convergence of the objective function value, predicted objective function value, design variables, and the Lagrange multipliers were also used. This sensitivity analysis was performed before the RQSEN system was built, and all of the calculations were performed by hand. Therefore we did not record all of the values of the design variables, Lagrange multipliers, and constraints.

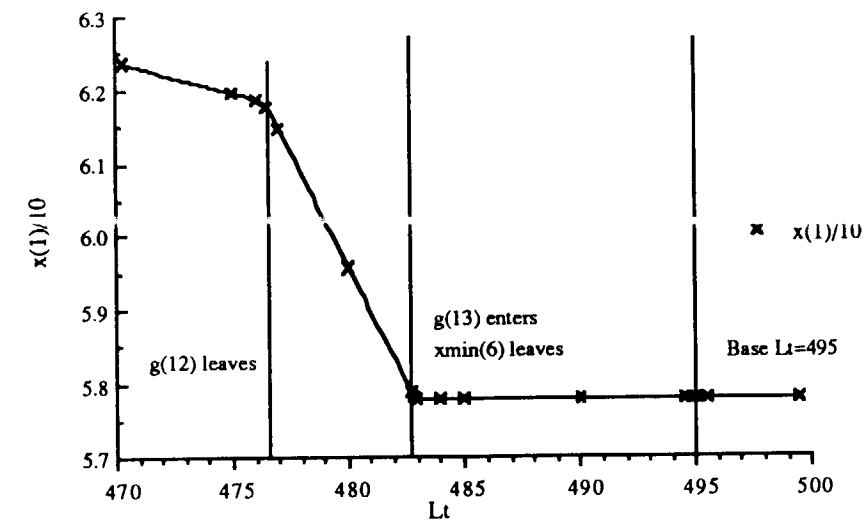
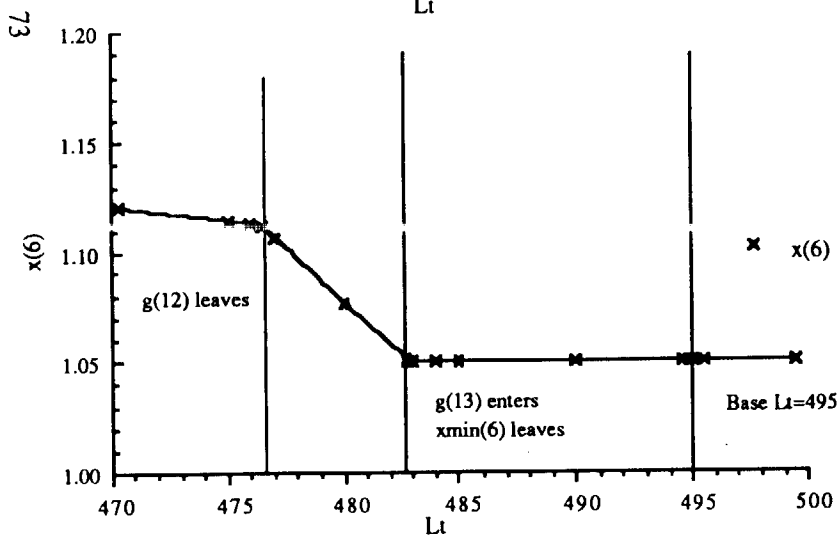
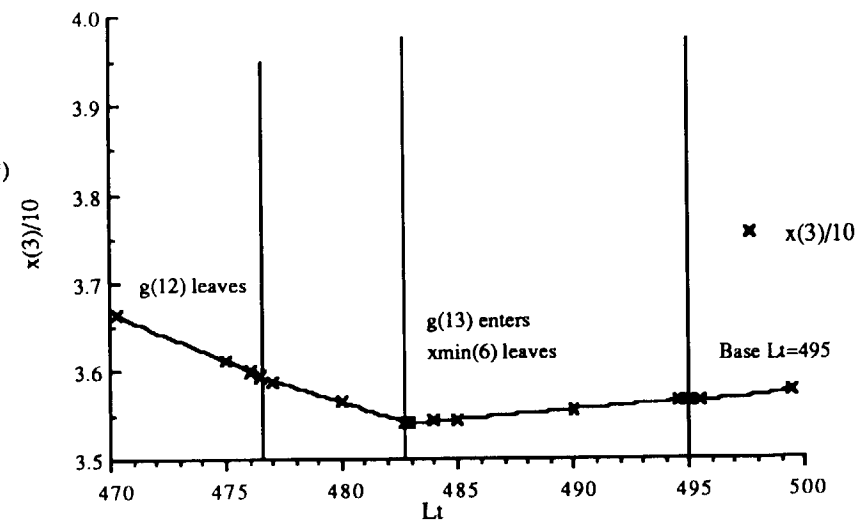
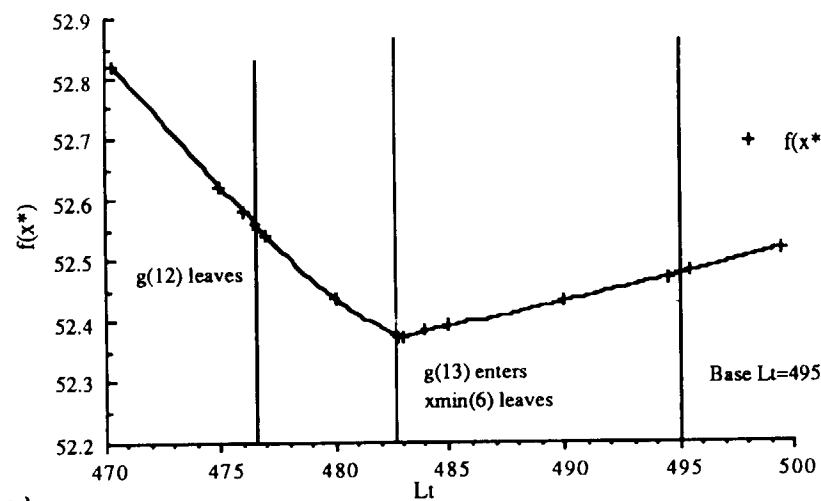
Figure 6.2(a) is a plot of the sensitivity of the optimal objective function for the bulkhead problem versus the parameter LT which was the length to the top brace. At $LT = 482.8$ we notice a discontinuity in the slope of the objective function. At this point we have encountered a Case 3 change in the active set. At $LT = 476.25$ there is a Case 2 change in the active set.

Figure 6.4 presents a plot of the value of constraint 13 (g_{13}) versus the value of LT . At $LT \approx 482.8$, the value of this constraint goes to zero and the constraint enters the active set.

Figure 6.3 presents a plot of the value of the Lagrange multiplier for constraint 12 versus LT . Notice that there is a discontinuity at $LT \approx 482.8$ this is due to a linear dependence of the constraint gradients of the active set. As LT moves from one side of 482.8 to the other, there will be a change in the active set of constraints with g_{13} replacing a constraint on the lower bound of x_6 .

Figure 6.2(d) presents a plot of the optimal value of design variable six. For $LT \geq 482.8$, the lower bound constraint is active. However, when $LT < 482.8$ the optimal value of this design variable is no longer on the bound. Figure 6.2 also presents plots of the behavior of x_1 and x_3 , we can also see discontinuities in these variables when the active set changes.

Figure 6.2 Sensitivity of the Bulkhead problem



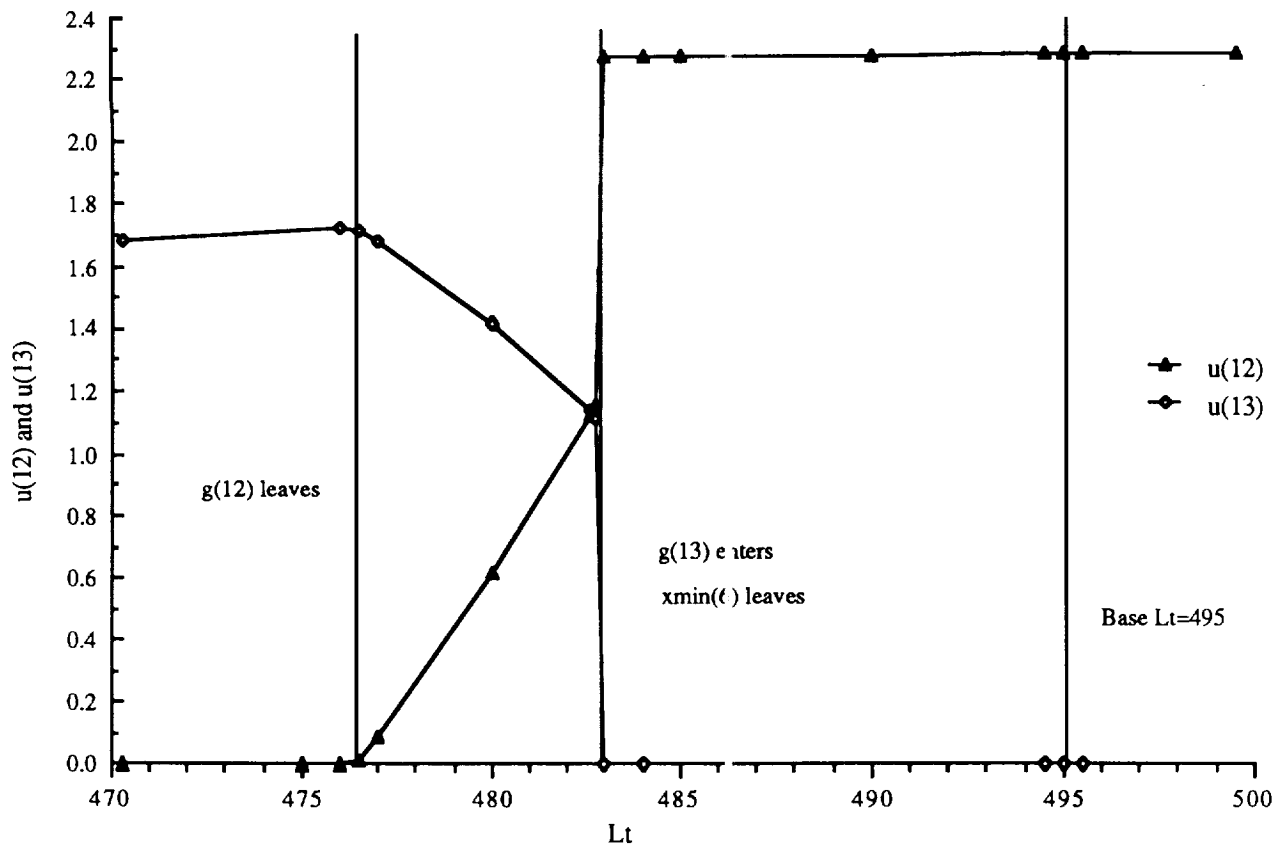


Figure 6.3 Plot of The Lagrange Multipliers for the Bulkhead problem

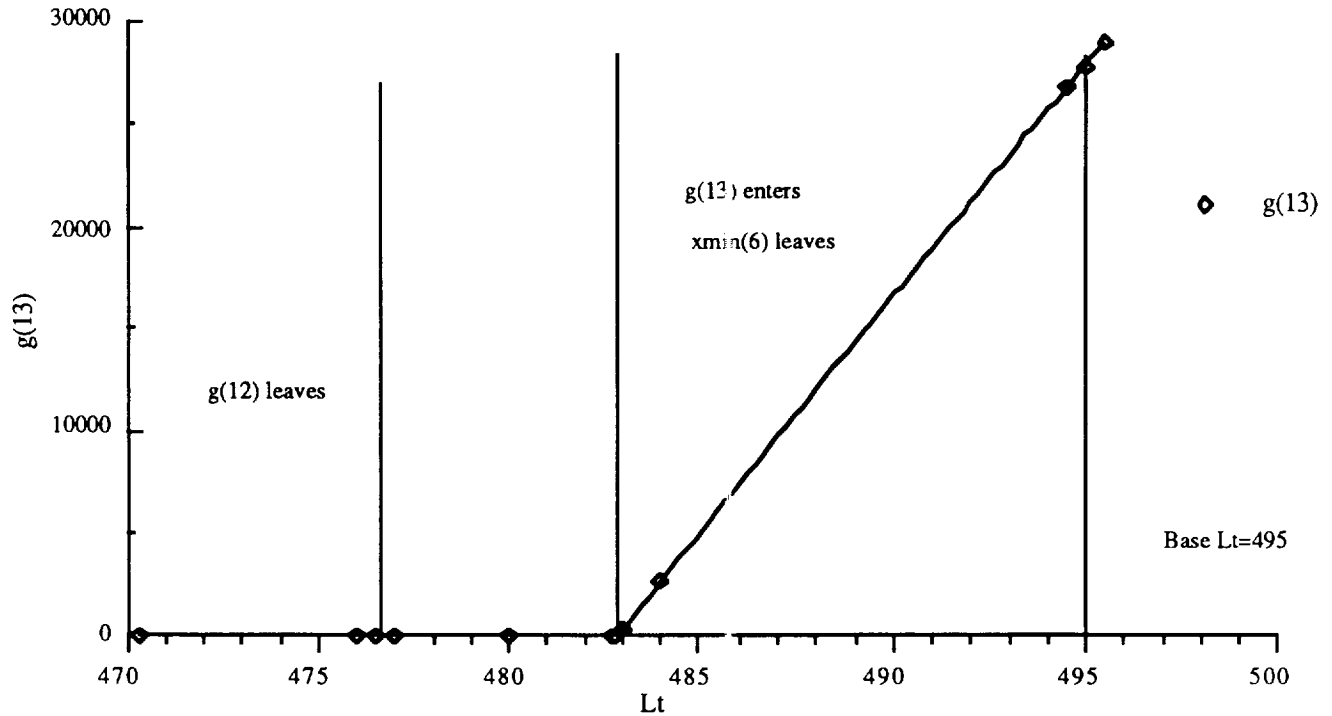


Figure 6.4 Plot of g_{13} versus Lt for the Bulkhead problem

For this problem there is a second change in the active set at $Lt \approx 476.25$ when g_{12} leaves the active set. As Lt moves through 476.25 there is not the same type of

discontinuity as we saw for Case 3. This represents Case 1, a constraint leaving the active set. We can see some of the characteristics of this case in the plots. In figure 6.2, the slope of the objective function is continuous at $LT = 476.25$. In figure 6.3, the value of the Lagrange multiplier of constraint g_{12} goes to zero signifying the constraint has left the active set. Finally, in figure 6.2 we can see that $\partial x_1/\partial p$, $\partial x_3/\partial p$, and $\partial x_6/\partial p$ are discontinuous at $LT = 476.25$.

A Case 4 change in the active set can be illustrated by the following simple one variable problem.

$$\begin{aligned} &\text{minimize } f(x) = x \\ &\text{Subject to: } g_1 = x - p_1 \geq 0 \\ &\quad \quad \quad g_2 = x - 2 \geq 0 \end{aligned}$$

When $p_1 = 1$ the optimum solution to this problem is $f^*=1.0$, $x^*=1.0$, and constraint g_1 is active. When p_1 is increased to $p_1=2.0$ constraint g_2 enters the active set. However, if p_1 is increased beyond 2.0 there is no feasible solution for this problem.

An engineering example of a case 4 change in the active set can be illustrated by the following example. Find the optimum air plane to fly a given mission, where the design variables may be the size of the wings, engine size, cruise altitude, etc. The design parameters might be; total cargo weight, runway length, air temperature at take-off, etc. Assume a constraint on take-off distance is applied,

$$\text{Take-off distance} \leq \text{Available Runway Length}$$

and a parameter sensitivity study of total cargo weight is performed. As the total cargo weight is increased, the take-off distance of the airplane increases, and the design variables of the airplane may also change. If the total cargo weight is increased beyond a certain limit it may become impossible for the aircraft to take-off at the given runway length. A parameter sensitivity analysis for values of the total cargo weight for values greater than this limit are meaningless because there will be no solution to the problem since the plane cannot take-off. Thus, for this problem when the runway length constraint enters the active set there will be a case 4 change in the active set causing no feasible solution to exist.

6.3. PROPOSED SOLUTIONS

This section will discuss some algorithms that can be used to obtain more accurate estimates of the parameter sensitivity after the active set of constraints has changed. The first subsection describes algorithms that can be used to calculate sensitivity derivatives when there are Case 1 or Case 2 change in the active set. The second subsection presents an example demonstrating how the algorithms presented in the previous section work. The

final subsection discusses how sensitivity derivatives can be calculated for Case 3 changes in the active set.

As we have mentioned previously, the sensitivity derivatives do not exist at points where the active set changes due to the discontinuities present in the optimum path. However, the sensitivity derivatives will exist in a directional sense. The proposed algorithms for dealing with changes in the active set are based on calculating directional derivatives.

Directional derivatives can be approximated as follows

$$\frac{df^*(\Delta p^+)}{dp} = \lim_{\Delta p \rightarrow 0^+} \left(\frac{f^*(p^0 + \Delta p) - f^*(p^0)}{\Delta p} \right) \quad (6.5)$$

Here $df^*(\Delta p^+)/dp$ indicates the rate of change of a function when a parameter is perturbed in the positive direction. When the base point for a sensitivity analysis is degenerate, we can approximate directional derivatives using the RQP sensitivity algorithm in the same way as we approximate other derivatives.

6.3.1 Dealing with Case 1 and 2

We begin this section by presenting the following example problem

$$\begin{aligned} \text{minimize: } f &= (x_1 + 1)^2 + (x_2 - 2)^2 & (6.6) \\ \text{subject to: } g_1 &= x_1 - p \geq 0 & (6.7) \\ g_2 &= 6 - 2x_1 - x_2 \geq 0 & (6.8) \end{aligned}$$

When $p = p_0 = 1$ as shown in figure 6.5, $f^* = 4$, $x^* = (1, 2)$. A sensitivity analysis indicates that as p is increased, $\partial x^*/\partial p = (1, 0)$. Notice, that as we increase p , eventually constraint g_2 will become active which is a Case 1 change in the active set. Increasing p from (x^*, p_1) will result in the optimum point moving along the intersection of the two constraints. The deflection algorithm is based on finding a constraint intersection and then calculating new values of $\partial x^*/\partial p$ and $\partial u^*/\partial p$ along the new active set.

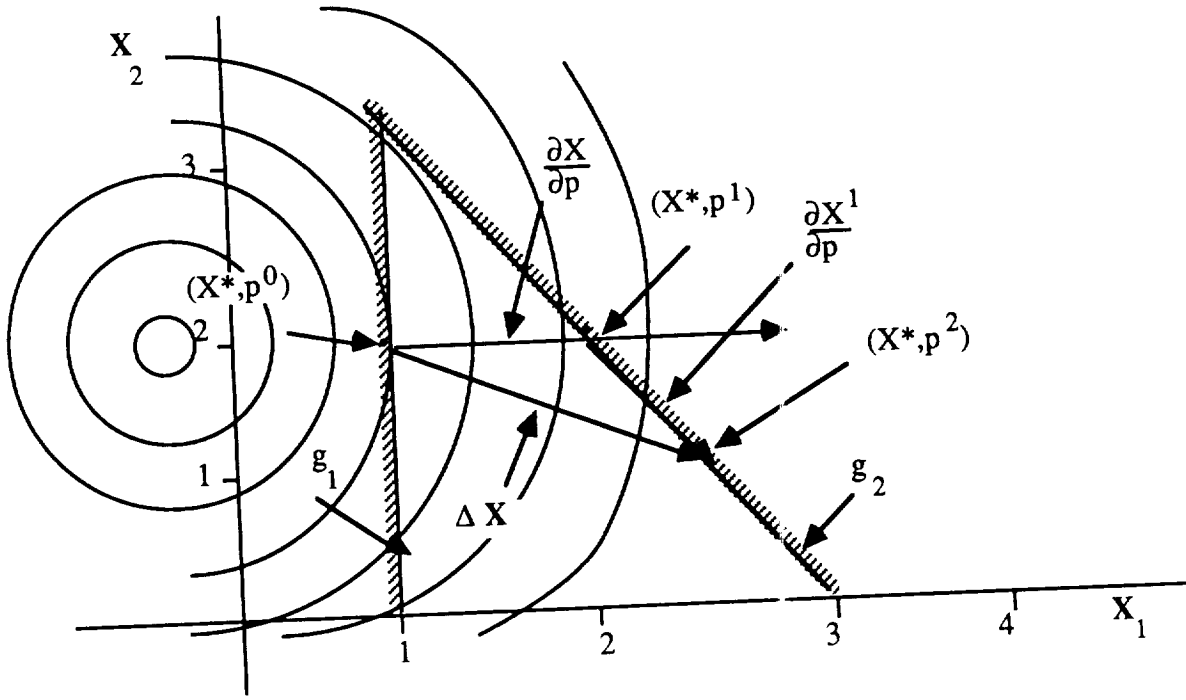


Figure 6.5 Deflection of the Sensitivity Search Direction

The proposed algorithm for dealing with constraints entering or leaving the active set is given by the following steps:

1. Determine $\partial \mathbf{x}^*/\partial p$ at the base point
2. Calculate Δp^1 for intersection of the constraint by finding the minimum Δp that causes a constraint to enter/leave the active set

$$\Delta p^1 = \min_j \left(\frac{g_j}{\frac{\partial g_j}{\partial p} + \frac{\partial g_j}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial p}} \right) \quad j \in \text{Active set} \quad (6.9)$$

$$\Delta p^1 = \min_j \left(\frac{u_j}{\left(\frac{\partial u_j}{\partial p} \right)} \right) \quad j \in \text{Active set} \quad (6.10)$$

3. Calculate $\partial \mathbf{x}^{*1}/\partial p$ with the active set updated to reflect the change indicated by step 2.
4. Calculate $\Delta \mathbf{x}$ and $\Delta \mathbf{u}$ by

$$\Delta \mathbf{x} = \frac{\partial \mathbf{x}^*}{\partial p} \Delta p^1 + \frac{\partial \mathbf{x}^{*1}}{\partial p} \Delta p^2 \quad (6.11)$$

$$\Delta \mathbf{u} = \frac{\partial \mathbf{u}^*}{\partial p} \Delta p^1 + \frac{\partial \mathbf{u}^{*1}}{\partial p} \Delta p^2 \quad (6.12)$$

where

$$\Delta p1 = \min (p - p0, 0) \quad (6.13)$$

$$\Delta p2 = \max (p - p1, 0) \quad (6.14)$$

5. Calculate the new value of the objective function by

$$f_{\text{new}} = f(\mathbf{x}^*, p0) + \Delta p \frac{\partial f}{\partial p} + \frac{\partial f}{\partial \mathbf{x}} \Delta \mathbf{x} \quad (6.15)$$

The first step in the above algorithm is to perform a sensitivity analysis to determine which direction the optimum will move in. In the second step, we attempt to identify the value of p for which the active set will change using equations 6.9 and 6.10. A new search direction is calculated in step three which includes the changed active set. In step four, a formula is provided which defines the proper value for $\Delta \mathbf{x}$ for before the constraint is encountered and after the constraint becomes active (inactive). This result is used in step 5 to predict a new value for the optimum at the new point.

In step three of the deflection algorithm, it is necessary to find the deflected search direction at the point where the new constraint enters the active set. This can be done by adding the newly violated constraint (or removing the constraint that is leaving the active set) to the Kuhn-Tucker sensitivity equations and then solving for $\partial \mathbf{x}^1 / \partial p$ and $\partial \mathbf{u}^1 / \partial p$. This computation can be performed efficiently by using matrix updating techniques as described in (Diewart 1984). In order to obtain a more accurate estimate of the sensitivities, the gradients of the active constraints can be re-evaluated at $\mathbf{x} = \mathbf{x}^1$, where \mathbf{x}^1 is the predicted intersection of the new constraint.

When using the deflection algorithm, a check should be made to assure that the constraint gradients are not linearly dependent. If the constraint gradients are linearly dependent then this procedure cannot be used to solve for the optimum sensitivities because there is no solution to the Kuhn-Tucker sensitivity equations. The procedure to be used when the constraint gradients are linearly dependent will be discussed in section 6.3.3.

The estimate obtained in step 5 is a linear extrapolation. A better estimate of the new value of the optimum can be made using a quadratic extrapolation. The formula for a quadratic extrapolation is (McKeown 1980, Fiacco 1983, Barthelemy and Sobieski 1983)

$$f^*_{\text{new}} = f(\mathbf{x}^*) + \Delta p_i \frac{df^*}{dp_i} + \frac{1}{2} \Delta p_i \frac{d^2 f^*}{dp_i^2} \Delta p_i \quad (6.16)$$

where

$$\frac{d^2 f^*}{dp_i^2} = \frac{\partial^2 L}{\partial p_i^2} + \left[\frac{\partial^2 L}{\partial p_i \partial \mathbf{x}^*} \right]^T \frac{\partial \mathbf{x}^*}{\partial p_i} - \frac{\partial \mathbf{u}^T}{\partial p_i} \frac{\partial \mathbf{g}}{\partial p_i} + \frac{\partial \mathbf{v}^T}{\partial p_i} \frac{\partial \mathbf{h}}{\partial p_i} \quad (6.17)$$

Using (6.16) to predict the new value of the objective function as p varies yields a more accurate estimate of the new optimal objective function until the active set changes. We should note that equation 6.17 is in terms of $\partial g/\partial p$ and $\partial h/\partial p$, and can be rewritten in the following form in terms of $\partial x^*/\partial p$

$$\frac{d^2 f^*}{dp_i^2} = \frac{\partial^2 L}{\partial p_i^2} + 2 \left[\frac{\partial^2 L}{\partial p_i \partial x^*} \right]^T \frac{\partial x^*}{\partial p_i} + \frac{\partial x^{*T}}{\partial p_i} \frac{\partial^2 L}{\partial x^{*2}} \frac{\partial x^*}{\partial p_i} \quad (6.18)$$

If we substitute equation 6.18 into equation 6.16 and write the equation in terms of Δp and Δx we will obtain

$$f^*_{\text{new}} = f_{1^{\text{st order}}} + \frac{1}{2} \left[\Delta p_i \frac{\partial^2 L}{\partial p_i^2} \Delta p_i + 2 \Delta p_i \frac{\partial^2 L}{\partial p_i \partial x^*} \Delta x + \Delta x^T \frac{\partial^2 L}{\partial x^{*2}} \Delta x \right] \quad (6.19)$$

where the first order approximation comes from equation 6.15. This form has the advantage of being able to predict the new value of the objective function at a new value of x and p . When the deflection algorithm is used to calculate the new location of the optimum then equation 6.19 can be used to make a second order estimate of the new value of the objective function.

As mentioned in section 6.1, a constraint leaving the active set (Case 2) is complementary to a constraint entering the active set (Case 1). It may be possible to predict df^*/dp when a constraint leaves the active set without calculating $\partial x^{1*}/\partial p$ and $\partial u^1/\partial p$ along the new active set. This could be beneficial for instances that only require estimates of $f^*(p)$. To predict the behavior of the optimum when a constraint leaves the active set we can use directional derivatives. Using the following formula (Fiacco 1983)

$$\frac{df^*}{dp} = \frac{\partial f}{\partial p} + \sum_{i=1}^{n_{eq}} v_i \frac{\partial h_i}{\partial p} - \sum_{i=1}^{n_{ineq}} u_i \frac{\partial g_i}{\partial p} \quad (6.20)$$

and the fact when an inequality constraint leaves the active set its Lagrange multiplier goes to zero, we can obtain an estimate of df^*/dp at the point where the change in the active set takes place. this is done by obtaining an estimate of $\partial u/\partial p$ from a sensitivity analysis at the base point using it be used to estimate when a constraint will leave the active set. A prediction of the df^*/dp when the active set changes can then be made from the following formula

$$\frac{df^*}{dp} = \frac{\partial f}{\partial p} + \sum_{i=1}^{n_{eq}} v_i^1 \frac{\partial h_i}{\partial p} - \sum_{i=1}^{n_{ineq}} u_i^1 \frac{\partial g_i}{\partial p} \quad (6.21)$$

where

$$v_i^1 = v_i^* + \frac{\partial v_i}{\partial p} \Delta p^1 \quad (6.22)$$

$$u_i^1 = u_i^* + \frac{\partial u_i}{\partial p} \Delta p^1 \quad (6.23)$$

When these values of u^1 and v^1 are used we obtain a new estimate for the value of the df^*/dp that will be valid when the value of p is increased or decreased past p^1 .

The deflection algorithm can be used to predict when a second change in the active set will take place. A prediction of when a second constraint may be added to or removed from the active set can be made by making a linear approximation using the following formulas

$$g_j^1 = g_j + \left[\frac{\partial g_j}{\partial p_i} + \frac{\partial g_j^T}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial p_i} \right] \Delta p_i^1 \quad j \notin \text{Active set} \quad (6.24)$$

$$u_j^1 = u_j + \left[\frac{\partial u_j}{\partial p_i} \right] \Delta p_i^1 \quad j \in \text{Active set} \quad (6.25)$$

and

$$\Delta p_i^2 = \min \left(\frac{g_j^1}{\frac{\partial g_j}{\partial p_i} + \frac{\partial g_j^T}{\partial \mathbf{x}^*} \frac{\partial \mathbf{x}^1}{\partial p_i}} \right) \quad j \notin \text{Active set} \quad (6.26)$$

$$\Delta p_i^2 = \min \left(\frac{u_j^1}{\left[\frac{\partial u_j^1}{\partial p_i} \right]} \right) \quad j \in \text{Active set} \quad (6.27)$$

where Δp_i^2 predicts the value of p_i that will cause the second constraint to enter (using eq. 6.26) or leave (using eq. 6.27) the active set, g_j^1 is a prediction of the value of the j^{th} constraint, and u_j^1 is the predicted value of the Lagrange multiplier when the active set changes. The predicted value for Δp^2 is calculated as a linear approximation of the value of the constraint in the $\partial \mathbf{x}^1 / \partial p$ direction. If the constraints are interrelated (i.e. are evaluated as a set) then it may be possible to use a more accurate estimate of $\partial g / \partial \mathbf{x}$ in formula 6.26, by predicting its new value by using the formula

$$\frac{\partial g}{\partial \mathbf{x}^*}(p^1) = \frac{\partial g}{\partial \mathbf{x}^*}(p^0) + \frac{\partial^2 g}{\partial \mathbf{x}^* \partial p_i} \Delta p_i^1 \quad (6.28)$$

When the constraints are interrelated $\partial^2 g / \partial \mathbf{x}^* \partial p$ can be evaluated when $\partial \nabla_{\mathbf{x}} L / \partial p$ is evaluated.

If the calculation of the gradient of a particular constraint is not expensive then the value of the new gradient may be used in formula 6.26.

The deflection algorithm and the variants that were introduced in this section will be effected by any nonlinearity that is present in the problem, particularly nonlinearity in the constraints. It should be emphasized that these sensitivities are only estimates of how the sensitivity will change.

6.3.2 An Example

This section presents an example problem to demonstrate how the deflection algorithm of section 6.2.1 performs. Example problem (5.6-8) will be used. Figure 6.5 shows the solution for $p = 1.0$, this example assumes that p is increased to $p = 2.5$. The exact value of the new optimum at $p = 2.5$ is $f^* = 13.25$, $X^* = (2.5, 1)$.

For $p = p_0 = 1.0$, the initial search direction for an increasing p is $\partial x^*/\partial p = (-1, 0)$ as shown in figure 6.5. Step 2 of our algorithm determines that constraint g_2 enters the active set when $p = p_1 = 2.0$. For values of p greater than p_1 the location of the optimum is along the intersection of constraints g_1 and g_2 . The new search direction along constraints g_1 and g_2 is determined from step three to be,

$$\frac{\partial x^1}{\partial p} = (-1, 2) \quad (6.29)$$

For $p = p^2 = 2.5$, which is greater than p_1 , the estimated Δx is composed of the sum of two vectors: one vector from (x^*, p_0) to (x^*, p_1) plus the vector from (x^*, p_1) to (x^*, p_2) . Thus by equation 6.11 for a $\Delta p = 1.5$

$$\Delta x = (1.5, -1) \quad (6.30)$$

Thus the estimate of the new location of the optimum for $p = 2.5$ becomes

$$x^*_{\text{new est}} = (2.5, 1) \quad (6.31)$$

Which is the true location of the new optimum for this problem. Without using the deflection algorithm we obtain the following estimate (by equation 2.9) of the optimum

$$x^*_{\text{new est}} = (2.5, 2) \quad (6.32)$$

By equation 2.7 and 2.8, the new value of the objective function will be

$$f_{KT1st} = 10 \quad (6.33)$$

Using equation 6.10, which takes into account the change in the active set, we get

$$f_{KTD1st} = 10 \quad (6.34)$$

Equation 6.16, which gives a second order estimate without taking into account the change in the active set, we obtain

$$f_{KT2nd} = 12.25 \quad (6.35)$$

Equation 6.19, which provides a second order estimate but does take into account the change in the active set, we obtain

$$f_{KTD2nd} = 13.25 \quad (6.36)$$

Using the deflection algorithm the predicted location of the optimum was in exact agreement with the true value. The linear predictions of f^*_{new} gave the same value because there was no component of the gradient of f in the x_2 direction. Using equation 6.16 for the second order estimate provided a better estimate of the new value of the objective function, but when equation 6.19 was used the exact value of the objective function was obtained. We should probably not expect results this good in more general optimization applications. However, we can expect better predictions of the location of the new optimum and the value of the objective function for small changes in the parameter when the active set changes.

6.3.3 Dealing with Case 3

Case 3 is the most difficult case to deal with for changes in the active set because when the active set changes, the Lagrange multipliers will be discontinuous and predicting the new active set as the parameter moves through the degenerate point is very difficult.

To deal with Case 3, we propose avoiding the singular point by reoptimizing the problem for values of p that are on either side of the singular point. Performing a sensitivity analysis at both points, use these sensitivities in a directional sense for p moving away from the singular point.

Reoptimizing the problem on either side of the singular point may be a difficult problem if the point is nearly singular. This may cause the algorithm being used in the reoptimization to fail or converge very slowly (Powell 1985, Bartholomew-Biggs 1986).

When using an algorithm such as RQOPT that updates an approximation of the Hessian of the Lagrangian, it may be unclear which values of the Lagrange multipliers to use if the Lagrange multipliers are not unique at the solution.

6.4. GENERATION OF TEST PROBLEMS

This section will describe a procedure that can be used to generate test problems that possess changes in the active set. The first section describes the generation of test problems with Case 1 and Case 2 changes in the active set. The second section describes the generation of test problems with Case 3 changes in the active set.

6.4.1 Test Problems for Case 1 and Case 2 Changes in the Active Set

Several test problems that exist in the literature possess Case 1 and 2 changes in the active set (See Schmit and Chang 1984, Vanderplaats and Yoshida 1985, Vanderplaats and Cai 1987). To generate new problems, active set changes for these two cases can be introduced into a test problem by adding constraints that are not active at the optimum but are violated for small changes in the parameters. The following is an outline of the steps that can be used to generate test problems where there are changes in the active set

Step 1 Generate a NLP Test Problem. One such method would be to use the Rosen and Suzuki (1965) procedure.

Step 2 Vary some problem parameters and find the path of the optimal solution with respect to p , say $x^*(p)$ from p^0 to p^2 .

Step 3 Choose a value of p^1 between p^0 and p^2 as the point where a constraint will enter the active set.

Step 4. Construct a new constraint such that

$$g_{\text{new}}(x^*(p^1), p^1) = 0$$

$$g_{\text{new}}(x^*(p^1 - \Delta p), p^1 - \Delta p) > 0$$

and the gradient of $g_{\text{new}}(x^*(p^1), p^1)$ is linearly independent of the gradients of the constraints that are already in the active set.

Step 5. Calculate the path of the new problem from p^1 to p^2

The above algorithm can be used to create test problems for Case 1 when a sensitivity analysis is conducted at $p = p^0$ and then p is perturbed to p^2 . The same test problem can be used as a Case 2 test problem, when the sensitivity analysis is performed at $p = p^2$ and p is moved to p^0 .

This procedure is illustrated in figure 6.6 which shows a graph of a two variable test problem along with the optimum $x^*(p^0)$. At the optimum, constraint g_1 is active. As the p increases from p^0 to p^1 the location of the optimum moves from $x^*(p^0)$ to $x^*(p^2)$, and constraint g_1 remains active. To introduce a change in the active set we can place an

additional constraint g_2 (shown in figure 6.6 (b)), that intersects the vector $\{\partial x(p^0)/\partial p\} \Delta p$, where $\Delta p = p^2 - p^0$. The value of p where the path to the optimum intersects the new constraint will be denoted as p^1 . By adding constraint g_2 , the optimum location $x^*(p^2)$ shown in figure 6.6 (b) will be different than without the constraint.

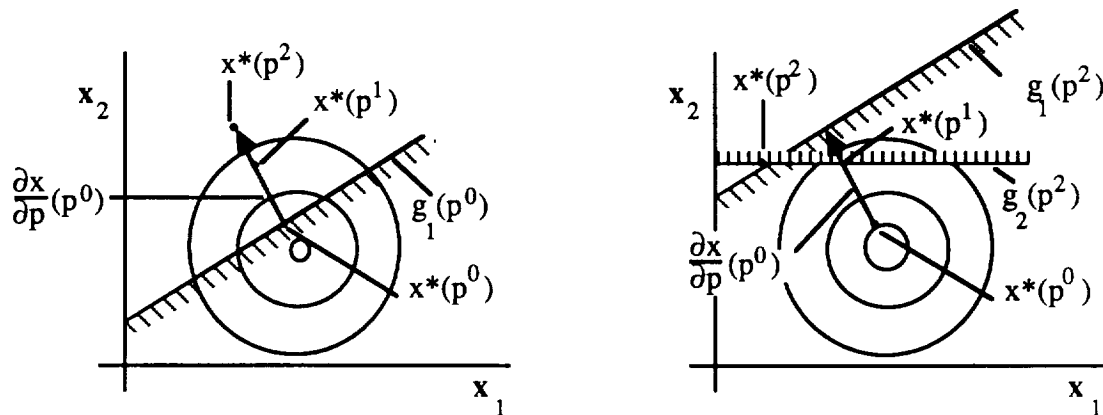


Figure 6.6 The creation of Test Problems with changes in the Active Set

6.4.2 Generation of Test Problems for Case 3

This section will describe the generation of test problems that have a Case 3 change in the active set. We first discuss problems that are in the literature. Then we will discuss two different algorithms that can be used to generate test problems of this type.

A survey of the literature revealed several test problems where the constraint gradients are linearly dependent when the active set changes (Case 3). Three such problems were found in articles by Bartholomew-Biggs (1986), Vanderplaats and Yoshida (1985), and Fiacco and Ghaemi (1982). It is suspected that problems discussed by Robertson and Gabriele (1987) and Barthelemy and Sobieski (1983) also possess Case 3 changes in the active set. Powell (1985) has studied the performance of RQP methods when the gradients of the constraints are linearly dependent. The test problems that were used by Powell can also be modified to be sensitivity test problems.

We can expect to find other test problems (Case 3) when we begin to study more engineering test problems. Many engineering optimization problems are fully constrained at the optimum. When a new constraint becomes active for a fully constrained problem we will either have a Case 3 change in the active set or loose the feasible region (Case 4).

To generate test problems for Case 3 changes in the active set the algorithm

described in section 6.4.1 can be used with the following modification. In step 4 when the new constraint is added to the active set it will have to be linearly dependent with the constraints in the active set. To accomplish this the gradient of the new constraint at $x(p^1)$ can be constructed as a linear combination of the gradients of the other active constraints. It may be possible with further work to control which constraint in the active set is replaced when the active set changes.

An alternative, less general procedure that can be used to create test problems with linearly dependent constraint gradients is illustrated for a two variable problem in figure 6.7. Figure 6.7 (a) shows a simple two variable optimization problem. The problem has an elliptic objective function and is subject to an equality constraint that changes as the parameter p changes. There are also variable bounds present. For $p = p^0$ the optimum is located at the intersection of the equality constraint and the variable bound $x_{2\max}$. When p is changed to $p = p^{\text{int}}$ the optimum is then located at the intersection of the equality constraint and both $x_{2\max}$ and $x_{1\min}$. At this point the gradients of the constraints are linearly dependent which causes the linear independence assumption of the Kuhn-Tucker conditions not to hold. When p changes further to $p = p^1$ as shown in figure 6.7 (b) the optimum is now located at the intersection of the equality constraint and $x_{1\min}$. Thus as p moves from p^0 to p^1 there will be a change in the active set with the constraint gradients being linearly dependent. This procedure can be generalized to more dimensions. The discontinuity in df^*/dp can be modified by varying the eccentricity of the ellipses.

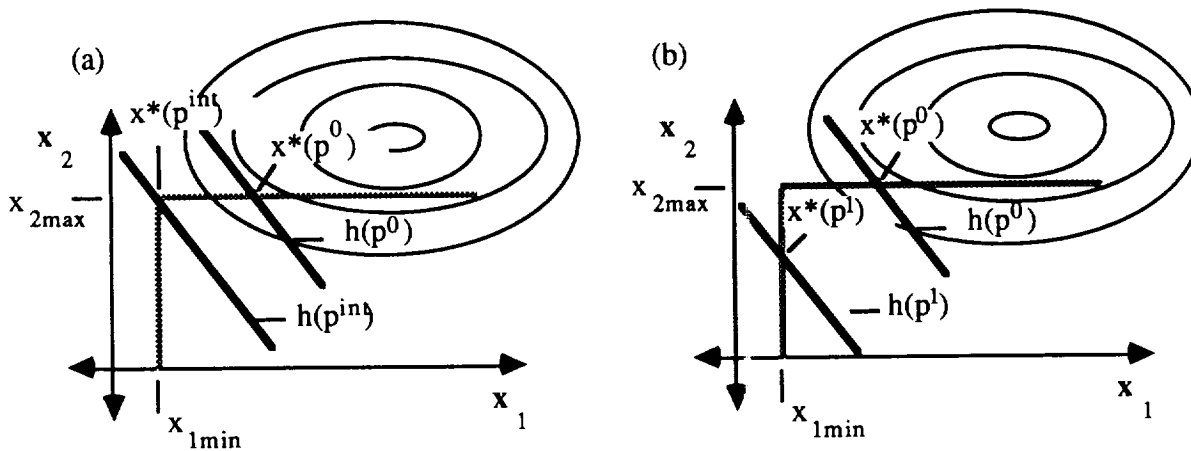


Figure 6.7 The Creation of Test Problems with Linear Dependencies

7. Conclusions and Recommendations for Future Research

7.1. CONCLUSIONS

In this work we have proposed a new method for estimating parameter sensitivity based on the Recursive Quadratic Programming method. The new method approximates the sensitivities using a differencing formula and can be shown to be equivalent to a modified Kuhn-Tucker method. The method appears to be very competitive with existing methods when measured in terms of the number of function evaluations required to calculate a parameter sensitivity. It does not require the calculation of second order derivatives, but uses the BFS method or SR1 method for developing an approximation to the Hessian of the Lagrangian.

The choice of the variable metric update (BFS or SR1) effects the amount of work required to solve the perturbed problem, because different updates provide Hessian approximations of differing accuracies. Different variable metric updates also effect the speed with which the RQP algorithm can solve the problem.

Initial testing of the algorithm against problems with known sensitivities has shown that the method can adequately estimate the derivatives. The central difference approximation seems to provide the best results, particularly when the Hessian approximation is updated during the RQP iterations at the perturbed point.

Using the RQP method to solve the optimization problem may be beneficial in many applications. This is because the RQP method has been shown to be one of the best general purpose algorithms for solving nonlinear programming problems. The use of variable metric updates allow the RQP method to solve problems where the Sequential Linear Programming (SLP) method fails. The RQP method can solve problems with very nonlinear constraints, and the RQP method performs the best when there are many active constraints at the optimum of the problem.

Chapter 6 has discussed potential problems occurring when there are changes in the set. Chapter 6 also presented some techniques that can be used to deal with these cases. We observed discontinuities of the sensitivity derivatives in Chapters 5 and 6 when there were changes in the active set of constraints. If we are using sensitivity derivatives to make extrapolations, we now know Case 3 will cause the largest discontinuities (i.e. step discontinuities in the Lagrange multipliers) in the derivatives, and Case 4 will cause no solution to the proposed constraints to exist. We have also shown that the discontinuities in $\partial x^*/\partial p$ and $\partial u^*/\partial p$, that occur when the active set changes, make our prediction of which constraint will enter the active set second very difficult without reoptimizing the

problem.

7.2. RECOMMENDATIONS FOR FUTURE RESEARCH

The next step in the testing of the new RQP sensitivity algorithm will be to expand the test set to include a larger variety of test problems. We will need to include problems that have more variables, and also problems that demonstrate the behavior associated with Case 3 and Case 4 changes in the active set. We will also need to expand the test set to include engineering test problems for which the Hessian of the Lagrangian is not readily available.

As we observed, the perturbation Δp can effect the accuracy of the derivatives that we calculated and we will have to investigate methods to improve the choice of Δp . In our initial testing we always let RQOPT use two iterations to solve the perturbed problem. Further tests are needed investigate the effectiveness of the algorithm when RQOPT is only allowed one iteration to solve the perturbed problem.

We observed that the Hessian of the Lagrangian improved if we allowed the approximation to be updated during the reoptimization. More experiments need to be conducted to find how to best update the Hessian approximation. We also need to find when we should switch from using two iterations to solve the perturbed problem to using one iteration to solve the perturbed problem. Further work is also needed to identify when the Hessian approximation has converged. If the Hessian approximation converges to the true Hessian of the Lagrangian then the Hessian approximation may be used in the Kuhn-Tucker sensitivity equations, however $\partial \nabla_x L / \partial p$ will still need to be calculated. Since $\nabla_x L \approx 0$ the calculation of $\partial \nabla_x L / \partial p$ may be subject to numerical noise and the RQP sensitivity algorithm may perform better than the Kuhn-Tucker method.

The initial testing of the SR1 update was very encouraging in terms of the convergence of the Hessian approximation to the True Hessian. However the SR1 update proved to be unstable when used for test problem 4. We will need to investigate some method that can be used to stabilize the SR1 update or find a set of rules that can be used that will only allow the SR1 update to be used when the updated Hessian approximation will be stable.

Using the RQP method to approximate the Hessian of the Lagrangian may be improved further if a Hybrid MOM/RQP algorithm is used to solve the original optimization problem. A Hybrid MOM/RQP algorithm would use the Method of Multipliers (or Augmented Lagrangian) algorithm for the first few stages, (build an approximation of the Hessian of the Lagrangian) then switch to using the RQP method. When the RQP method is used the approximation of the Hessian of the Lagrangian from

then MOM method can be used as the initial Hessian approximation, this should help the RQP method quickly solve the problem, and also obtain a good approximation of the Hessian of the Lagrangian.

In summation we have seen that the new RQP sensitivity algorithm can find parameter sensitivities. We still need to investigate if this method will be superior to the Kuhn - Tucker method for general problems. Section 3.2.2 demonstrated that there will be a trade off with regarding the number of function evaluations required by the Kuhn-Tucker method versus the RQP method. We will have to conduct more experiments to study the general accuracy that can be expected from the RQP sensitivity algorithm.

Using first order extrapolations can provide unsatisfactory results when the functions are nonlinear with respect to p . This situation is illustrated in Figure 7.1, which shows the effect of variations in p_3 on x_1^* . On this plot a linear and quadratic extrapolation are presented as well as the actual values of the optimum of $x_1^*(p_3)$. It can be seen that the linear extrapolation does not provide a good estimate of the new value of x_1^* when p_3 changes, however the quadratic approximation provides an accurate estimate of $x_1^*(p_3)$ up until the point where the active set changes.

If good estimates of the second derivatives can be found then more accurate estimates of the behavior of the optimum can be made by using second order extrapolations.

There are few available methods to calculate second derivatives of the optimum with respect to parameter variation but it is possible to predict d^2f^*/dp_i^2 for some problems. However the only published algorithm that was found for calculating $\partial^2x^*/\partial p_i^2$ requires third order derivatives, which are seldom available for engineering problems. Thus, an algorithm based on the central differencing variants of the RQP sensitivity algorithm is proposed that can be used to calculate second derivatives.

When using the central difference approximation an estimate of the second derivatives can be calculated from

$$\frac{d^2y(p_i)}{dp_i^2} = \frac{y^+ - 2y + y^-}{(\Delta p_i)^2} \quad (7.1)$$

Where y can represent f^* , x^* , u^* , or g^* , and the estimates of $f^{+,-}$, $x^{+,-}$, $u^{+,-}$, and $g^{+,-}$ calculated in steps 2 and 4 of the RQP sensitivity algorithm are substituted appropriately in (7.1). When using a central differencing approximation this option can be effective for indicating when curvature is present, but may not be able to accurately predict the true value of the second order information. It should be noted that this procedure may have the most

difficulty in predicting $\partial^2 u(p_i)/\partial p_i^2$ because the RQP algorithm produces more accurate estimates of $f^{+,*}$ and $x^{+,*}$ than $u^{+,*}$.

Second derivatives might also be useful for predicting when constraints enter the active set. A prediction of when a constraint enters the active set can be identified using the following model of the behavior of the constraints that are not in the active set

$$g_{\text{new}} = g(x^*, p) + \frac{dg}{dp_i} \Delta p_i + \frac{1}{2} \frac{d^2 g}{dp_i^2} (\Delta p_i)^2 \quad (7.2)$$

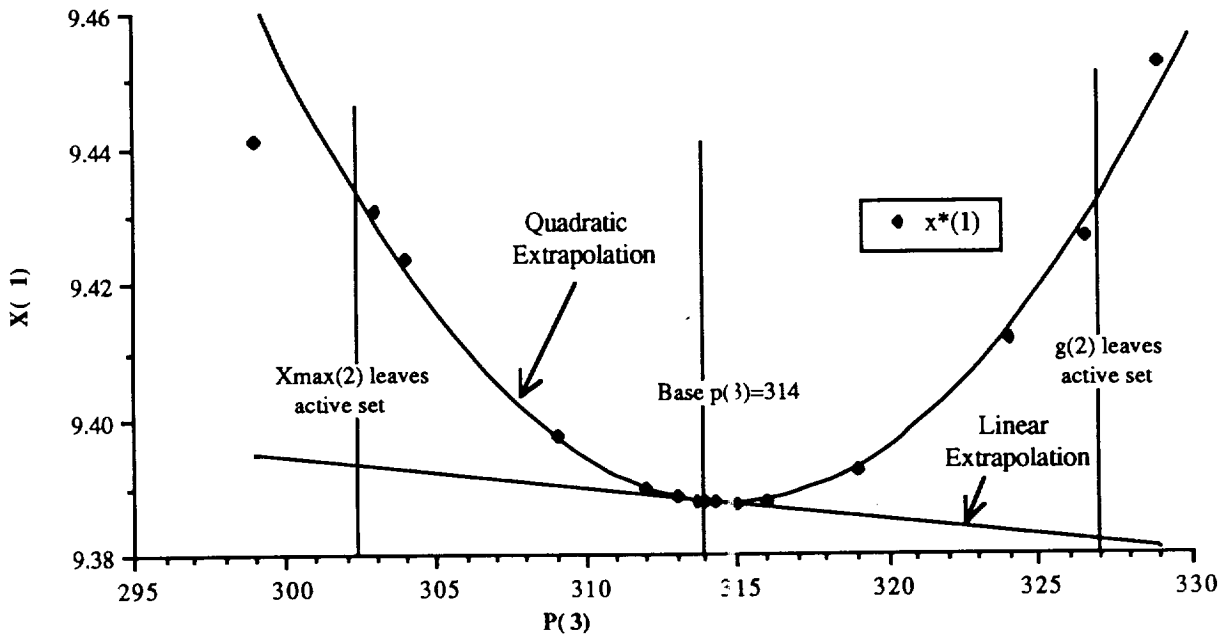


Figure 7.1 A Comparison of Quadratic to Linear Extrapolations for the Sensitivity Approximations.

As a final note to this report we will discuss using the RQP method and the RQP sensitivity algorithm in multilevel decomposition algorithms. Because multilevel decomposition algorithms solve the same subproblems for different values of system level parameters, we expect that as the multilevel decomposition algorithm converges (after the proper active set has been identified) that the true Hessian of Lagrangian in the subsystem will not change very drastically. If the RQP method is used to solve the subsystem level optimizations and perform the sensitivity analysis, we expect that the Hessian approximation will converge to the true Hessian of the Lagrangian as the multilevel decomposition converges. If we use the approximation of the Hessian of the Lagrangian from the previous subsystem optimization as the initial Hessian approximation for the next iteration, a better approximation of the Hessian of the Lagrangian should be obtained when we solve the new subsystem problem.

The above discussion implies that using the RQP method in conjunction with the

multilevel decomposition method may mean that on the first few system level iterations the sensitivities calculated at the subsystem level may be inaccurate, but the accuracy of the sensitivity derivatives will improve after each iteration at the system level. In our opinion this behavior is suitable for use with multilevel decomposition since far from the optimum it is often not advantageous (or necessary) to perform exact line searches or have exact values of the gradients. However, as the solution is approached, we need to obtain more accurate derivatives and perform more exact line searches to obtain acceptable convergence.

Appendix 1

Test problems

This section will present a discussion of the test problems that were used in the initial testing. Selected plots of the optimum sensitivity are provided to show how the optimum varies when the parameter is perturbed. The plots will also demonstrate how changes in the active set effect the optimum sensitivity.

On each plot that is presented, the base value of the parameter will be indicated by a vertical line indicating the value at which the sensitivity analysis was performed. Active set changes will be indicated by a vertical line and a label indicating the constraint that enters or leaves the active set when the parameter is perturbed from the base point. On the plots of the optimum objective function vs the parameter, a linear extrapolation will be indicated by a line showing the predicted value of the objective function using equation 2.8.

The plots of the behavior of the optimum can be used as a tool when diagnosing the behavior of various algorithms used to predict parameter sensitivities. Nonlinearity in the paths $f^*(x^*, p)$, $x^*(p)$, and $u^*(p)$ can be seen in these plots, this nonlinearity can be used to explain why (or how far) linear extrapolations are valid. The discontinuities in the sensitivities that occur when the active set changes can also be seen and we can use these discontinuities to establish regions where the extrapolations are valid.

Problem 1:

$$\text{Minimize } f(\mathbf{x}) = (x_1 - p_1)^2 + (x_2 - p_3)^2$$

$$\text{subject to: } g_1(\mathbf{x}) = 2x_1 - x_2^2 - p_2 \geq 0$$

$$g_2(\mathbf{x}) = p_3^2 - 0.8x_1^2 - 2x_2 \geq 0$$

$$\text{Variable bounds } \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \mathbf{x} \leq \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$$\text{Starting Point for Optimization } \mathbf{x}^0 = (0,3) \quad \mathbf{p}^0 = (3,1,3)$$

$$\text{Optimum Point: } f(\mathbf{x}^*(\mathbf{p}^0)) = 1.25 \quad \mathbf{x}^*(\mathbf{p}^0) = (2.5, 2.0)$$

$$\text{Both constraints are active: } \mathbf{u}^*(\mathbf{p}^0) = (0.3, 0.4)$$

$$\text{Hessian of the Lagrangian } \mathbf{H} = \begin{bmatrix} 2.64 & 0 \\ 0 & 2.6 \end{bmatrix}$$

Sensitivity derivatives

$$\frac{df}{dp_1} = 1.0 \quad \frac{\partial \mathbf{x}^*}{\partial p_1} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_1} = \begin{bmatrix} -0.2 \\ 0.4 \end{bmatrix} \quad \text{varied } p_1 \text{ from 1.5 to 4.8}$$

$$\frac{df}{dp_2} = -0.3 \quad \frac{\partial \mathbf{x}^*}{\partial p_2} = \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_2} = \begin{bmatrix} -0.1304 \\ 0.0008 \end{bmatrix} \quad \text{varied } p_2 \text{ from -3.0 to 1.5}$$

$$\frac{df}{dp_3} = 0.4 \quad \frac{\partial \mathbf{x}^*}{\partial p_3} = \begin{bmatrix} 1.2 \\ 0.6 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_3} = \begin{bmatrix} .4048 \\ -0.5896 \end{bmatrix} \quad \text{varied } p_3 \text{ from 2.2 to 3.95}$$

Special features: Hessian matrix is available. Quadratic objective function and quadratic constraints. Active set changes are introduced for large changes in the parameters. Problem is fully constrained at the optimum. Plots of the behavior of the optimum are presented in figures A.1 and A.2.

Constraint g_1 leaves the active set when $p_1 \geq 4.5$, and constraint g_2 leaves the active set when $p_1 \leq 2.0$.

Constraint g_2 leaves the active set when $p_3 \geq 3.888$, and constraint g_1 leaves the active set when $p_3 \leq 2.4322$. Since $\mathbf{u}^*(p_3)$ is nonlinear a linear prediction of when the constraint will leave the active set will not be accurate for large variations in the parameter.

Figure A.1 Sensitivity of Problem 1 with respect to P(1)

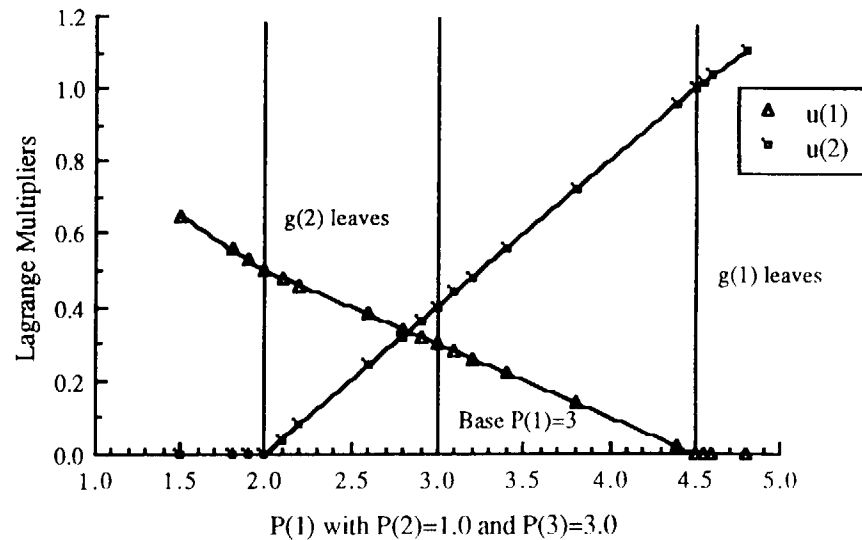
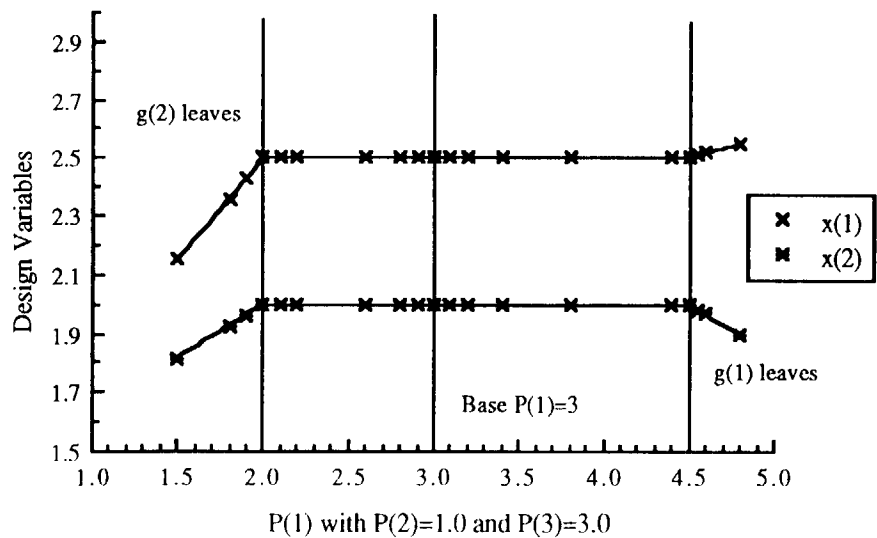
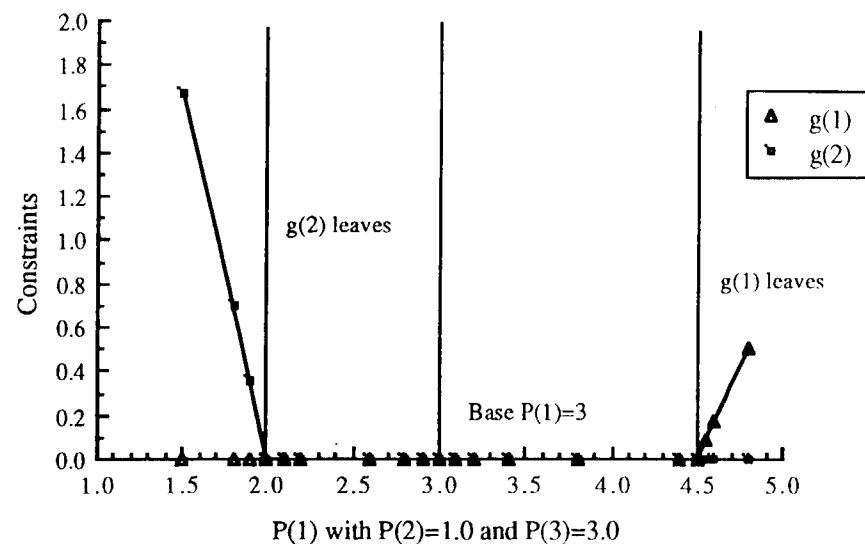
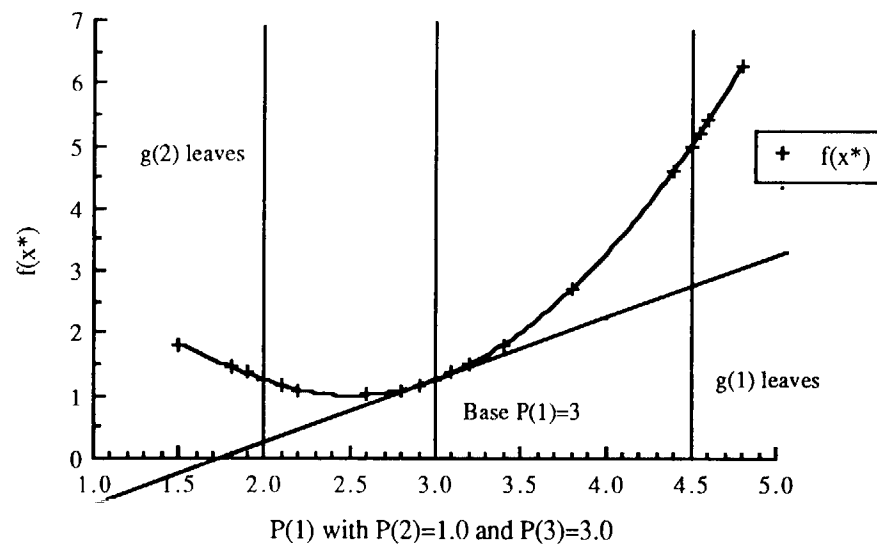
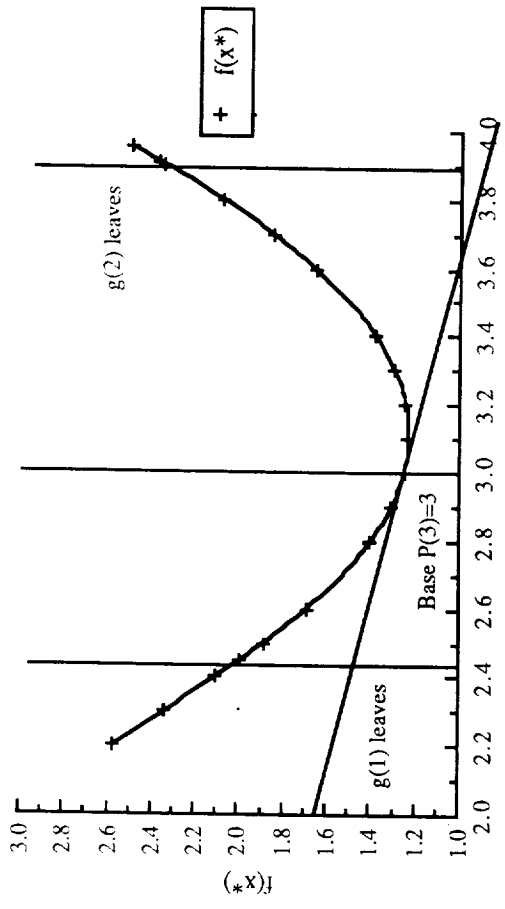
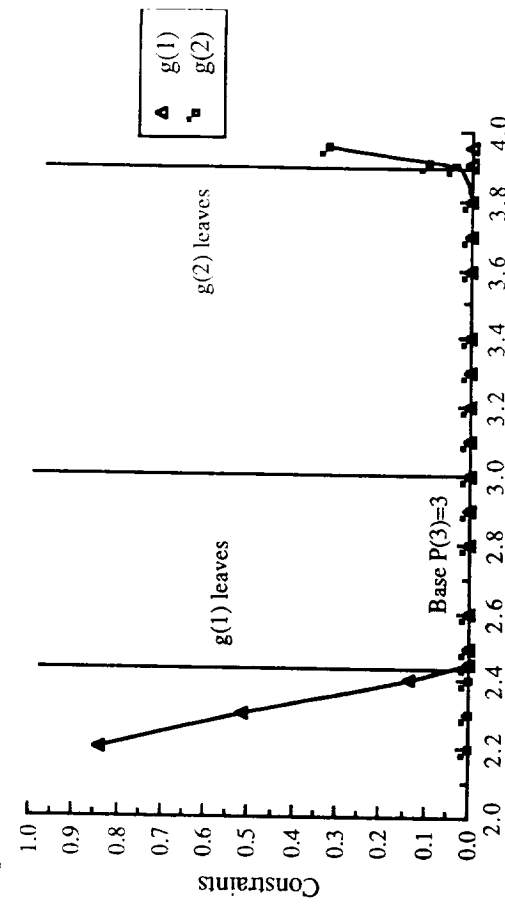


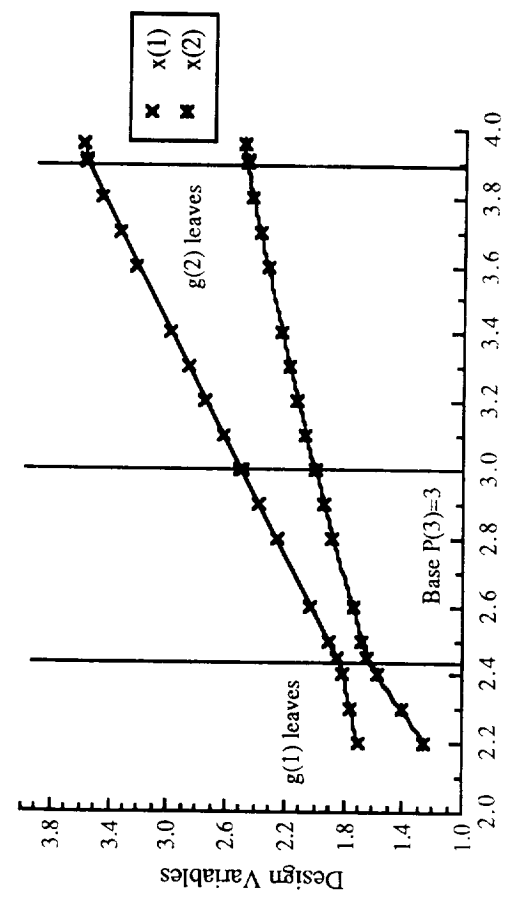
Figure A.2 Sensitivity with respect to $p(3)$ for Problem 1



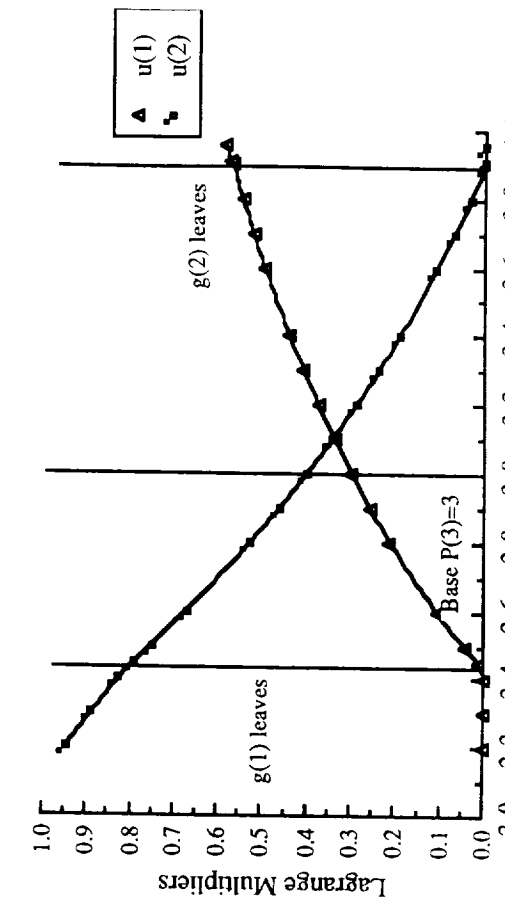
$P(3)$ with $P(2)=1.0$ and $P(1)=3.0$



$P(3)$ with $P(2)=1.0$ and $P(1)=3.0$



$P(3)$ with $P(2)=1.0$ and $P(1)=3.0$



$P(3)$ with $P(2)=1.0$ and $P(1)=3.0$

Problem 2:

$$\text{Minimize } f(\mathbf{x}) = 0.5\mathbf{x}^T \begin{bmatrix} 5 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \end{bmatrix} \mathbf{x} + \mathbf{x}^T \begin{bmatrix} 5 \\ 5p_1 \\ 5 \end{bmatrix}$$

$$\text{subject to: } g_1(\mathbf{x}) = p_1x_1 + x_2 + x_3 - p_2 \geq 0$$

$$g_2(\mathbf{x}) = x_1 + 2x_2 + 3x_3 - 4.7 - p_1 \geq 0$$

$$\text{Variable bounds } \begin{bmatrix} -10 \\ -10 \\ -10 \end{bmatrix} \leq \mathbf{x} \leq \begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix}$$

$$\text{Starting Point for Optimization } \mathbf{x}^0 = (1.1, 1.2, 1.3) \quad \mathbf{p}^0 = (1, 3)$$

$$\text{Optimum Point: } f(\mathbf{x}^*(\mathbf{p}^0)) = 25.5 \quad \mathbf{x}^*(\mathbf{p}^0) = (1.0, 1.0, 1.0)$$

$$g_1 \text{ active at the optimum: } \mathbf{u}^*(\mathbf{p}^0) = (12.0, 0.0)$$

$$\text{Hessian of the Lagrangian } \mathbf{H} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \end{bmatrix}$$

Sensitivity derivatives

$$\frac{df}{dp_1} = -7.0 \quad \frac{\partial \mathbf{x}^*}{\partial p_1} = \begin{bmatrix} 2.0833333 \\ -2.1666666 \\ -0.9166666 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_1} = \begin{bmatrix} -4.6666666 \\ 0.0 \end{bmatrix} \quad \text{varied } p_1 \text{ from } 0.8 \text{ to } 1.2$$

$$\frac{df}{dp_2} = 12.0 \quad \frac{\partial \mathbf{x}^*}{\partial p_2} = \begin{bmatrix} 0.3333333 \\ 0.3333333 \\ 0.3333333 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_2} = \begin{bmatrix} 2.3333333 \\ 0 \end{bmatrix} \quad \text{varied } p_2 \text{ from } 2.7 \text{ to } 3.2$$

Special features: Hessian matrix is available. Quadratic objective function and linear constraints. Parameter 2 is a right hand side perturbation.

A plot of the behavior of the optimum with respect to p_1 is presented in figure A.3. For $p_1 \geq 1.04889$, constraint g_2 enters the active set. We can see a change in sign of d^2f^*/dp_1^2 from a region of positive curvature to a

region of negative curvature. We can also see the discontinuity of the slope of $\partial \mathbf{x}^*/\partial p$ and $\partial \mathbf{u}^*/\partial p$ when the active set changes.

A plot of the behavior of the optimum with respect to p_2 is presented in figure A.4. For $p_1 \leq 0.95$, constraint g_2 enters the active set.. Again we can see the discontinuity in $\partial \mathbf{x}^*/\partial p_2$ and $\partial \mathbf{u}^*/\partial p_2$.

Figure A.3 Sensitivity with respect to $p(1)$ for Problem 2

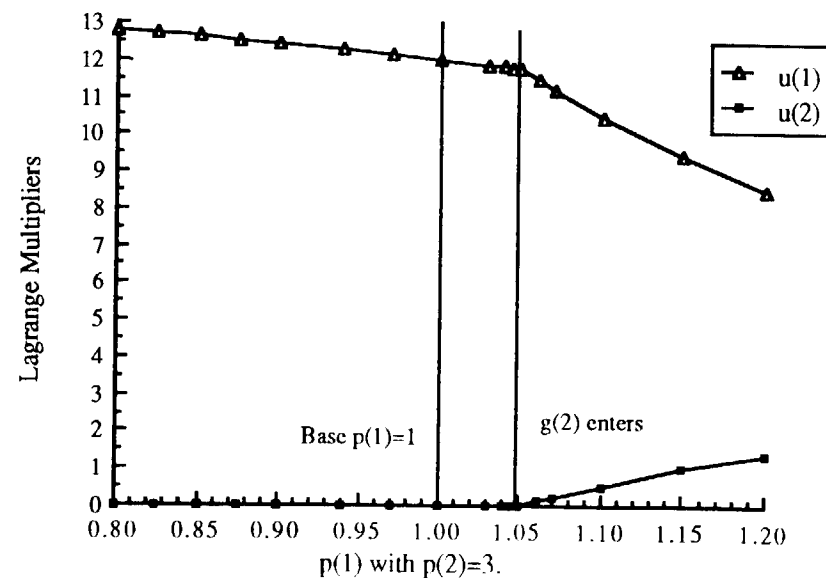
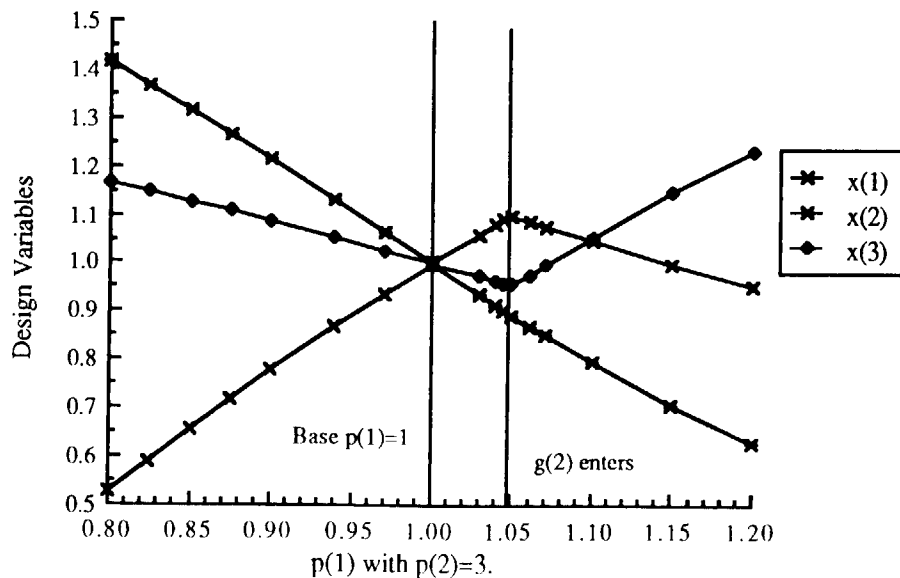
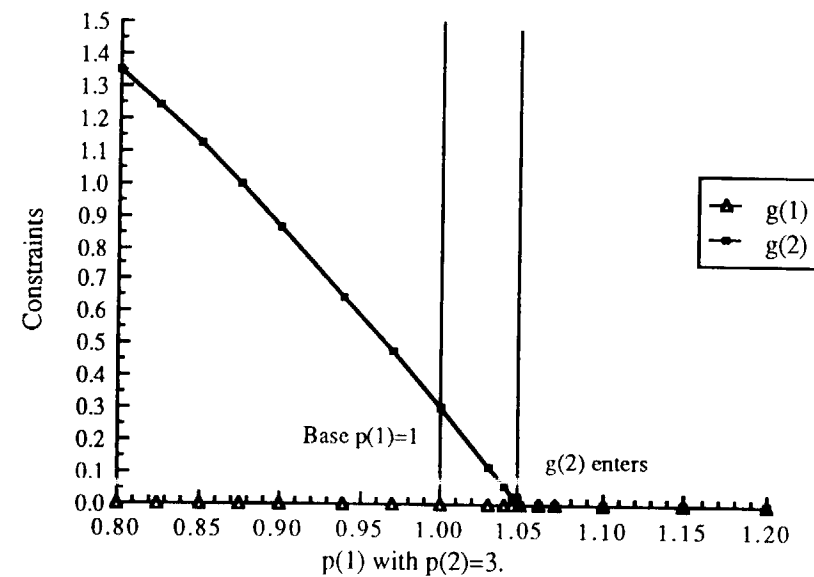
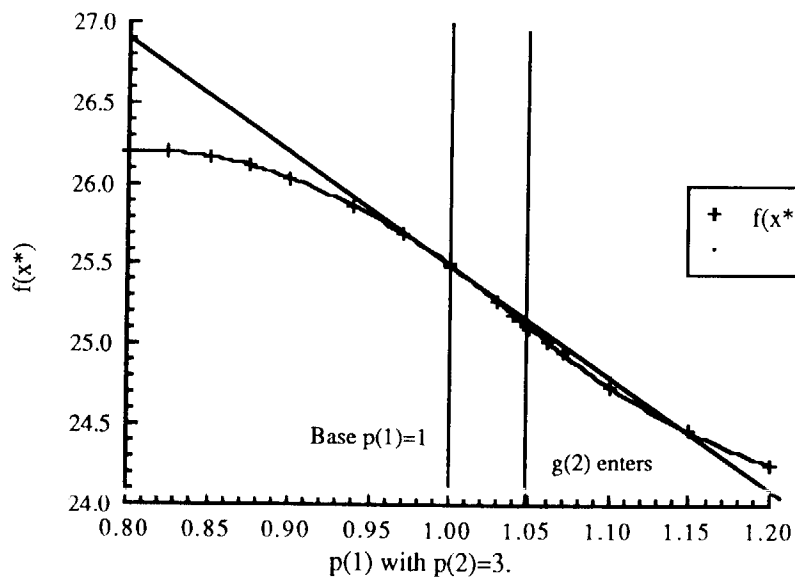
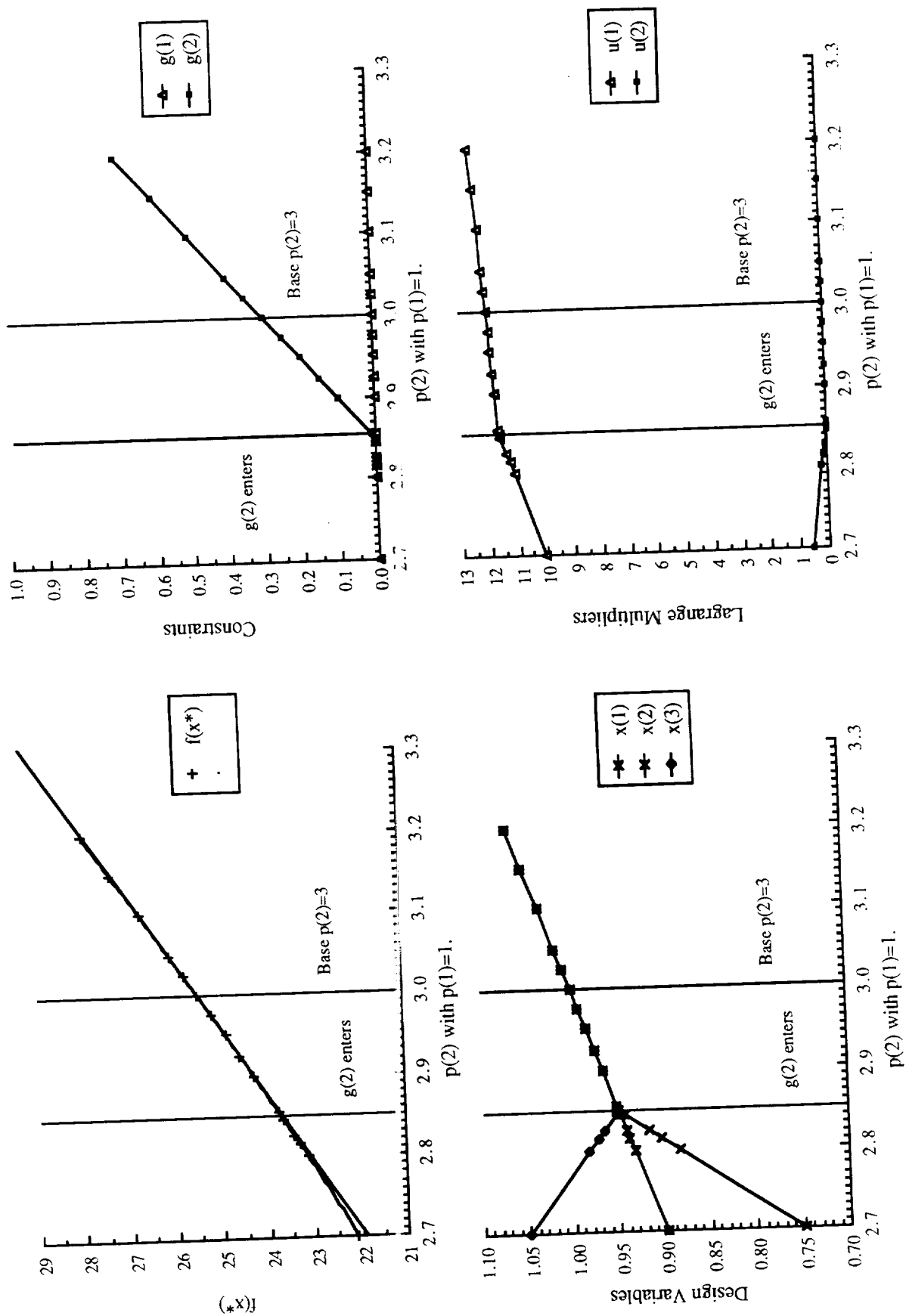


Figure A.4 Sensitivity with respect to $p(2)$ for Problem 2



Problem 3 (Common name Rosen and Suzuki Test Problem):

$$\text{Minimize } f(\mathbf{x}) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 + 100$$

$$\text{subject to: } g_1(\mathbf{x}) = (-x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4)/8 + p_1 \geq 0$$

$$g_2(\mathbf{x}) = (-x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4)/10 + p_2 \geq 0$$

$$g_3(\mathbf{x}) = (-2x_1^2 - x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4)/5 + p_3 \geq 0$$

$$\text{Variable bounds } \begin{bmatrix} -10 \\ -10 \\ -10 \\ -10 \end{bmatrix} \leq \mathbf{x} \leq \begin{bmatrix} 20 \\ 20 \\ 20 \\ 20 \end{bmatrix}$$

$$\text{Starting Point for Optimization } \mathbf{x}^0 = (0.0, 0.0, 0.0, 0.0) \quad \mathbf{p}^0 = (1, 1, 1)$$

$$\text{Optimum Point: } f(\mathbf{x}^*(\mathbf{p}^0)) = 56.0 \quad \mathbf{x}^*(\mathbf{p}^0) = (0.0, 1.0, 2.0, -1.0)$$

$$g_1 \text{ and } g_3 \text{ active: } \quad \mathbf{u}^*(\mathbf{p}^0) = (0.0, 0.0, 10.0)$$

$$\text{Hessian of the Lagrangian } \mathbf{H} = \begin{bmatrix} 12 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Sensitivity derivatives

$$\frac{df}{dp_1} = -8.0 \quad \frac{\partial \mathbf{x}^*}{\partial p_1} = \begin{bmatrix} -1.1471984 \\ -0.33428030 \\ -0.2279022 \\ -3.5403608 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_1} = \begin{bmatrix} -67.3428300 \\ 0.0 \\ 55.4605800 \end{bmatrix} \quad \text{varied } p_1 \text{ from 0.86 to 1.12}$$

$$\frac{df}{dp_2} = 0.0 \quad \frac{\partial \mathbf{x}^*}{\partial p_2} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_2} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad \text{varied } p_2 \text{ from 0.85 to 1.15}$$

$$\frac{df}{dp_3} = -10.0 \quad \frac{\partial \mathbf{x}^*}{\partial p_3} = \begin{bmatrix} 1.3057920 \\ 0.5460588 \\ -1.0541310 \\ -2.3741690 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_3} = \begin{bmatrix} -67.3428300 \\ 0.0 \\ 55.4605800 \end{bmatrix} \quad \text{varied } p_3 \text{ from 0.86 to 1.12}$$

Special features: Hessian matrix is available. Quadratic objective function and quadratic constraints. Problem has been used in the other studies of optimal design, the parameters that we are studying are right hand side perturbations.

Figure A.5 and A.6 present plots of the variation of the optimum with respect to p_1 . The following changes in the active set are introduced: when $p_1 \leq .855$, constraint g_3 leaves the active set; when $p_1 \geq 1.057$, constraint g_2 enters the active set; when $p_1 \geq 1.078$, constraint g_1 leaves the active set. The plot shows that $f^*(p_1)$ is nonlinear. It is difficult to see the discontinuity in $\partial x^*/\partial p_1$ but we can clearly see the discontinuity in $\partial u^*/\partial p_1$. Figure A.7 presents a close up view of the behavior of x_3 and x_4 . We can see that x_3 is a nonlinear and a piecewise continuous function of p_1 . The discontinuities take place when the active set changes. With the resolution in figure A.5 this behavior was very difficult to see, however with the enlarged view this becomes easy to see.

A plot of the behavior of the optimum with respect to p_2 is presented in figure A.8. When $p_2 \leq 0.9$, constraint g_2 enters the active set. In figure A.8 we can see a sharp discontinuity in $\partial u/\partial p$ and we also can see that large errors would be introduced if the value of the objective function was extrapolated from the base point to after the active set changed.

Figure A.5 Sensitivity with respect to $p(1)$ for Problem 3

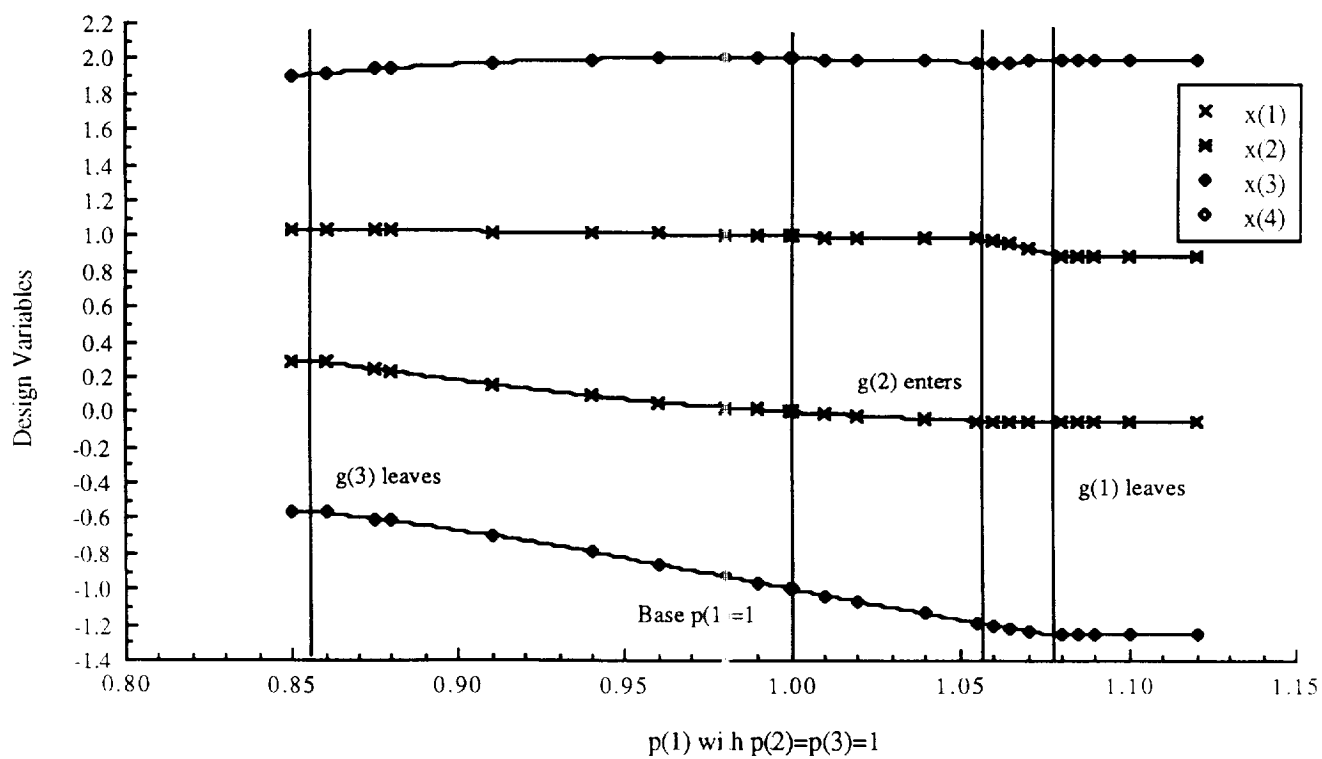
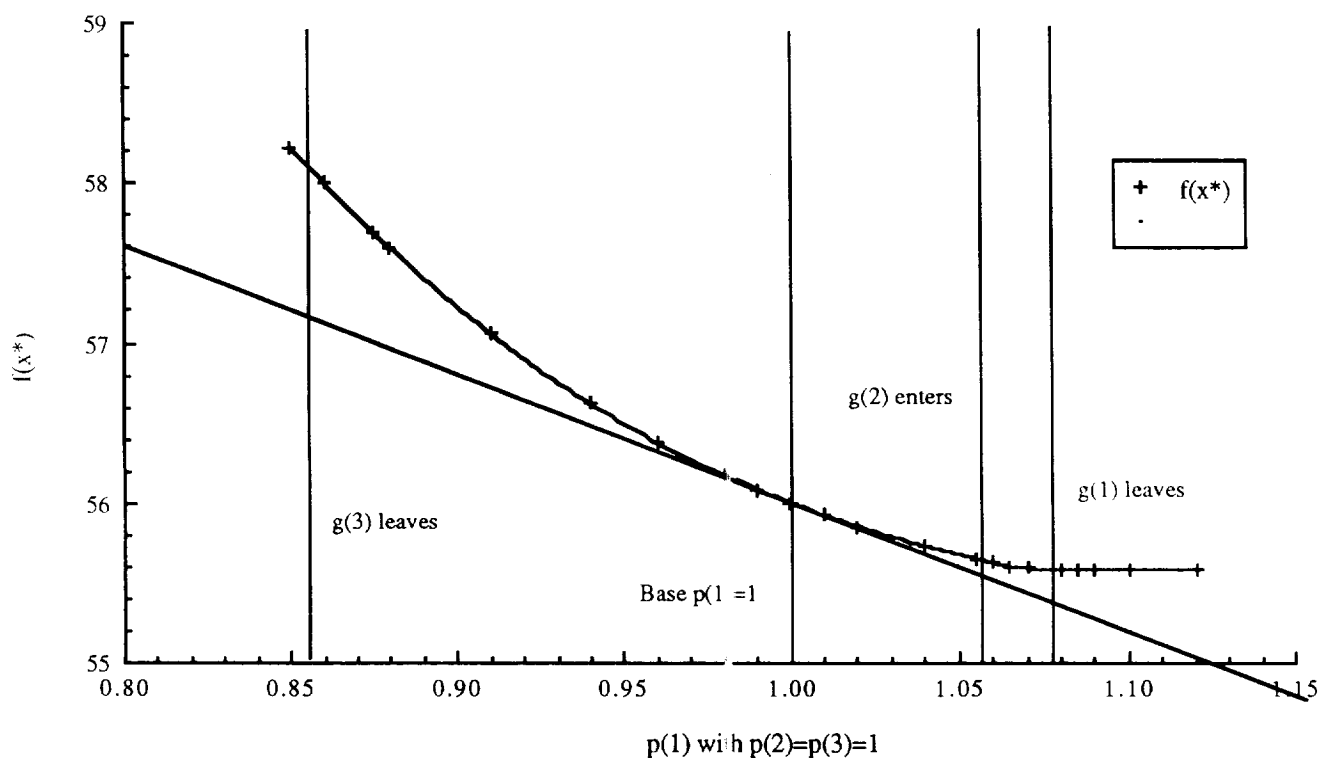
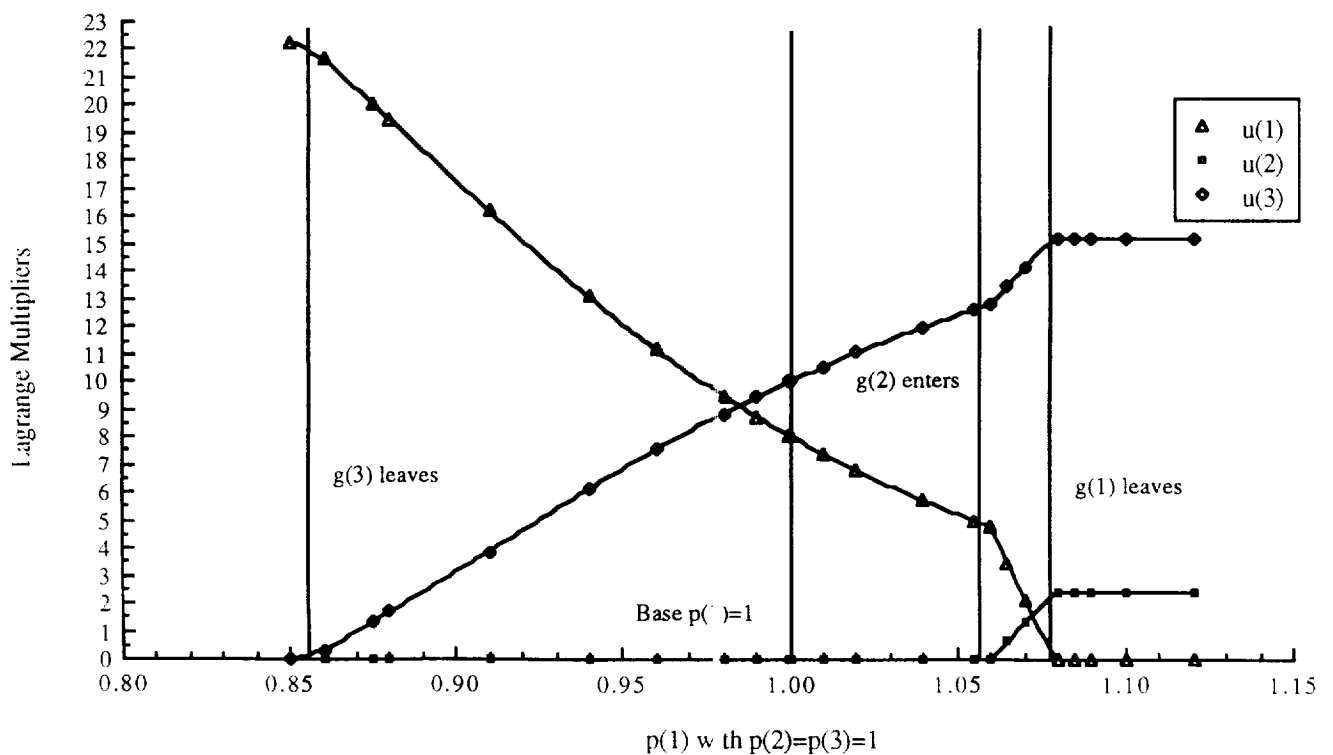
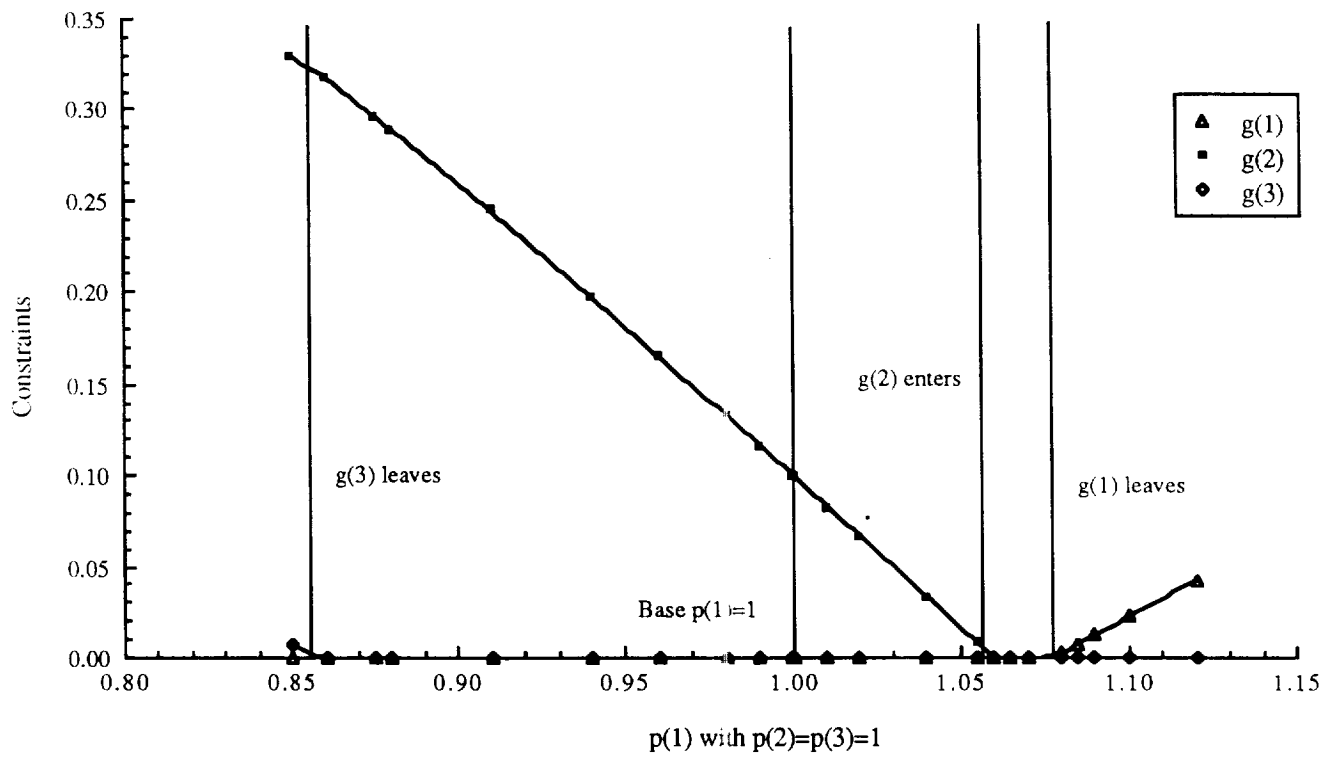


Figure A.6 Sensitivity with respect to $p(1)$ for Problem 3 (cont)



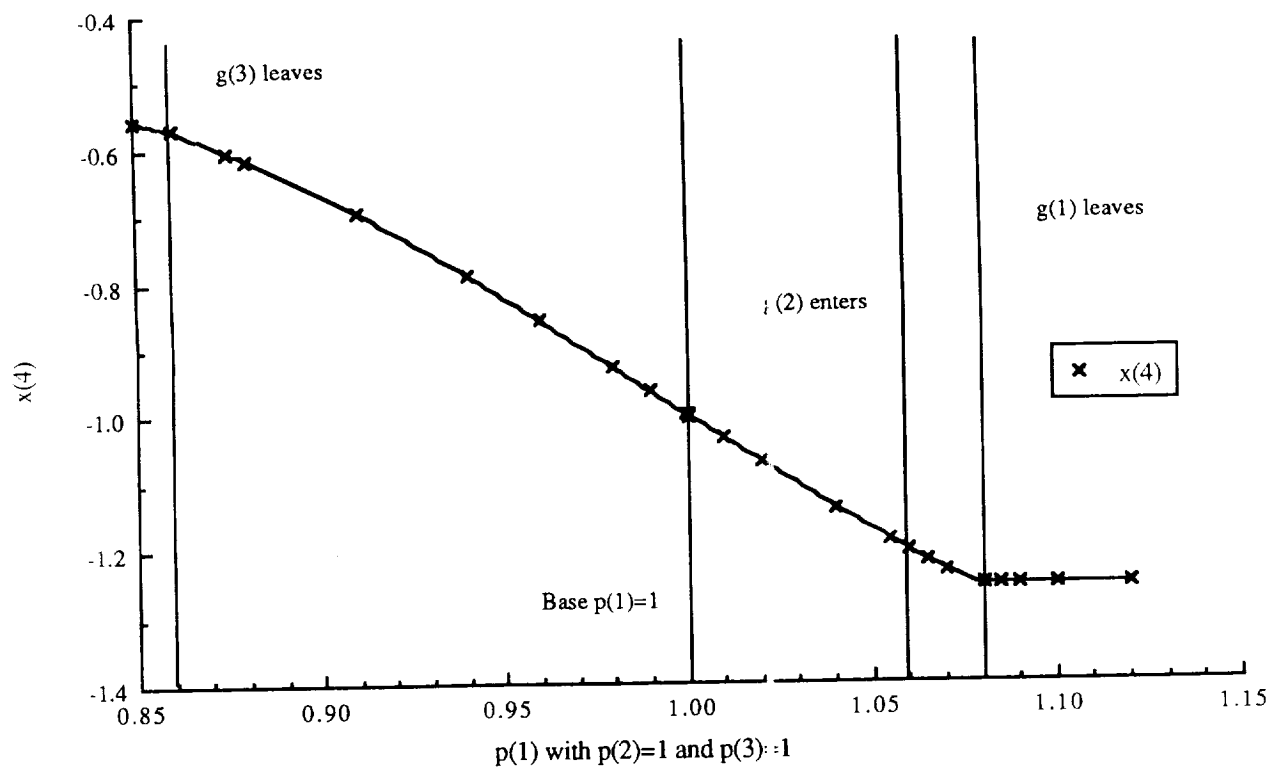
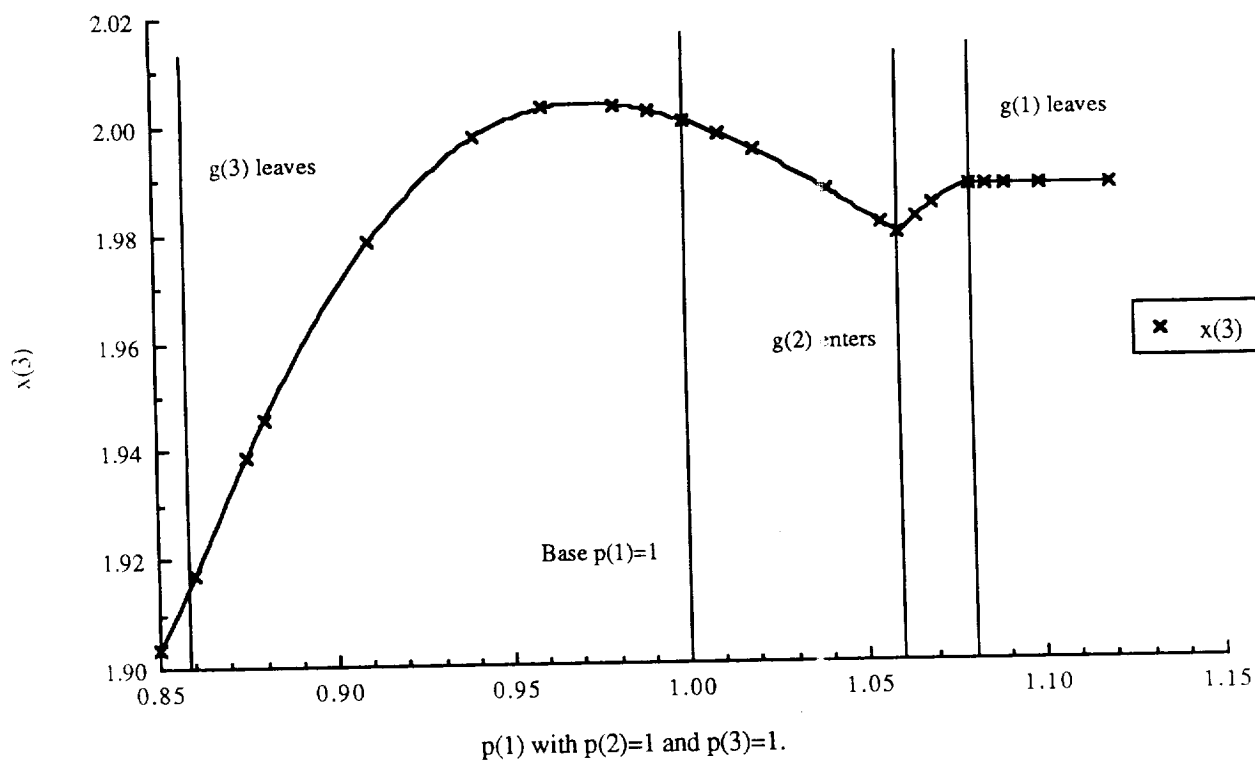
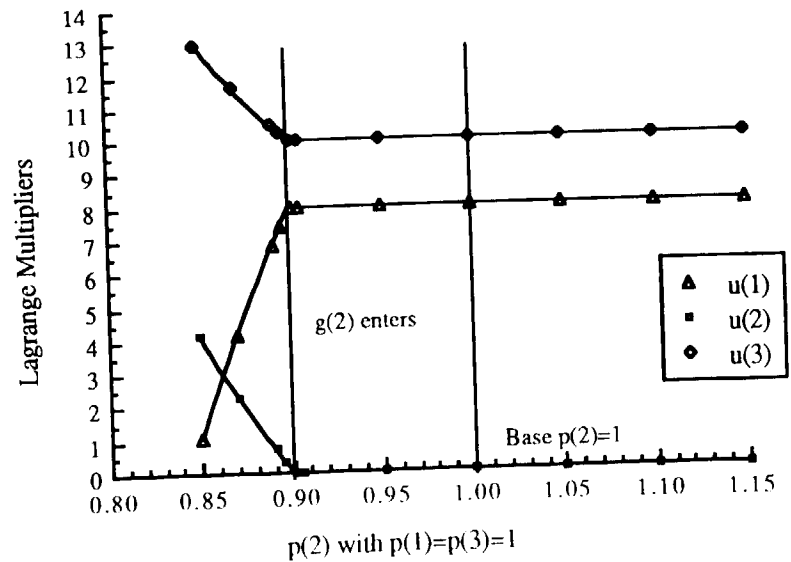
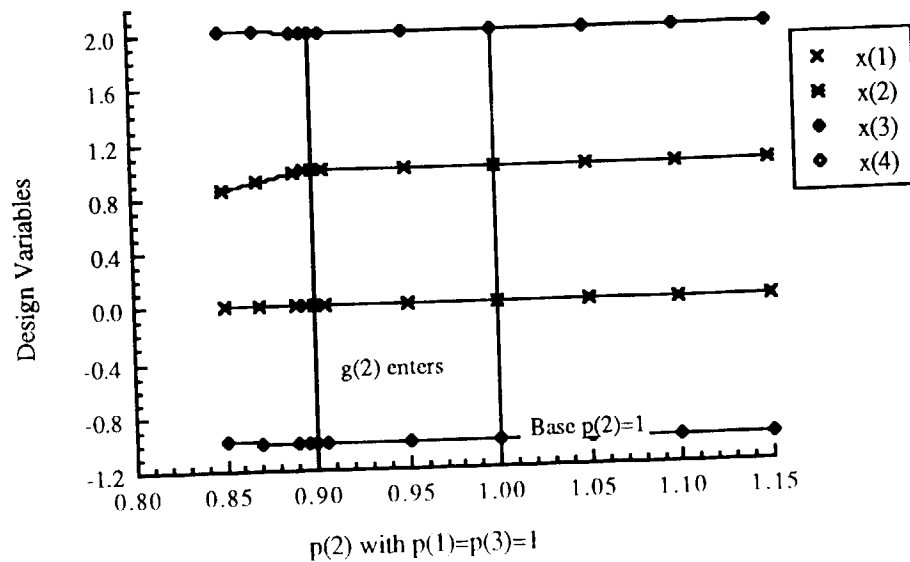
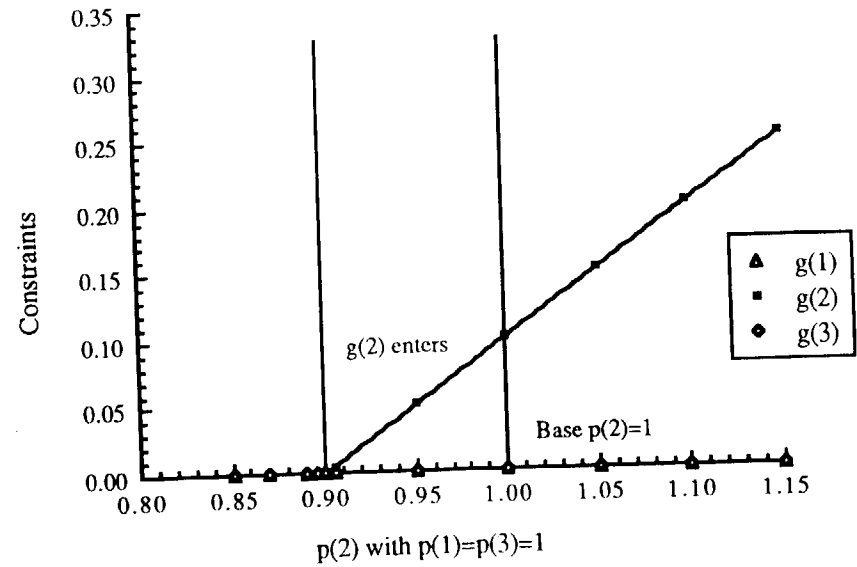
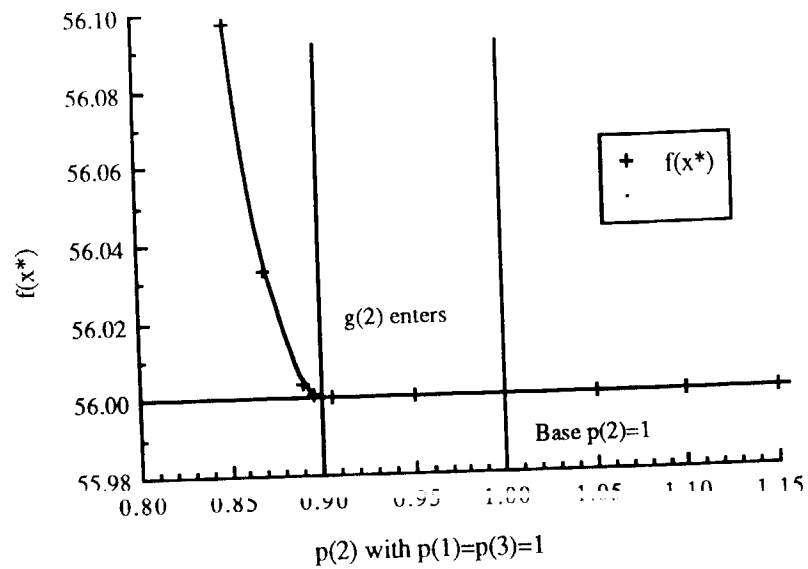


Figure A.7 Plots of the Sensitivity of $x(3)$ and $x(4)$ for problem 3 $p(1)$

Figure A.8 Sensitivity with respect to $p(2)$ for Problem 3



Problem 4:

$$\text{Minimize } f(\mathbf{x}) = \sum_{j=1}^5 e_j x_j + \sum_{i=1}^5 \sum_{j=1}^5 c_{ij} x_i x_j + \sum_{i=1}^5 d_i x_i^3$$

$$\text{subject to: } g_i(\mathbf{x}) = \sum_{j=1}^5 a_{ij} x_j - b_i \geq 0 \quad i=1,10$$

Where the values of a_{ij} , b_i , c_{ij} , d_i , e_i are constants that can be found in (Coville 1969, Himmelblau 1972, Eason and Fenton 1974, or Sandgren 1977)

$$\text{Variable bounds } \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq \mathbf{x} \leq \begin{bmatrix} 20 \\ 20 \\ 20 \\ 20 \\ 20 \end{bmatrix}$$

Starting Point for Optimization $\mathbf{x}^0 = (0., 0., 0., 0., 1.)$ $\mathbf{p}^0 = (b_3, b_5, b_6, b_9) = (-.25, -4., -1., 5.)$

Optimum Point: $f(\mathbf{x}^*(\mathbf{p}^0)) = -3.234866708$

$$\mathbf{x}^*(\mathbf{p}^0) = (0.3, 0.3334676, 0.4, 0.4283101, 0.2239649)$$

g_3 , g_5 , g_6 , and g_9 are active:

$$\mathbf{u}^*(\mathbf{p}^0) = (0., 0., 0.517404, 0., 0.306111, 1.183954594, 0., 0., 0.010390, 0.)$$

$$\text{Hessian of the Lagrangian } \mathbf{H} = \begin{bmatrix} 6.72 & -4.0 & -2.0 & 6.4 & -2.0 \\ -4.0 & 9.4006 & -1.2 & -6.2 & 6.4 \\ -2.0 & -1.2 & 4.4 & -1.2 & -2.0 \\ 6.4 & -6.2 & -1.2 & 9.3418 & -4.0 \\ -2.0 & 6.4 & -2.0 & -4.0 & 6.2688 \end{bmatrix}$$

Sensitivity derivatives

$$\frac{df}{dp_1} = .517404 \quad \frac{\partial \mathbf{x}^*}{\partial p_1} = \begin{bmatrix} -0.4 \\ 0.097300 \\ -0.2 \\ 0.286025 \\ -0.067740 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_1} = \begin{bmatrix} 0.407099 \\ -0.056624 \\ 0.347303 \\ 0.019084 \end{bmatrix} \quad \text{varied } p_1 \text{ from } -0.39 \text{ to } -.019$$

$$\frac{df}{dp_2} = .306111 \quad \frac{\partial \mathbf{x}^*}{\partial p_2} = \begin{bmatrix} 0.0 \\ -0.147119 \\ -0.0 \\ -.049166 \\ 0.09817 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_2} = \begin{bmatrix} -0.056624 \\ .0505155 \\ -.061564 \\ 0.0024016 \end{bmatrix} \quad \text{varied } p_2 \text{ from } -4.5 \text{ to } -3.5$$

$$\frac{df}{dp_3} = 1.18395 \quad \frac{\partial \mathbf{x}^*}{\partial p_3} = \begin{bmatrix} -0.2 \\ 0.094082 \\ -0.35 \\ 0.228817 \\ 0.029313 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_3} = \begin{bmatrix} .34303 \\ -0.061564 \\ 0.720695 \\ 0.159647 \end{bmatrix} \quad \text{varied } p_3 \text{ from } -1.2 \text{ to } -0.8$$

$$\frac{df}{dp_4} = .010390 \quad \frac{\partial \mathbf{x}^*}{\partial p_4} = \begin{bmatrix} 0.0 \\ -0.039759 \\ 0.0 \\ 0.0761408 \\ .15499104 \end{bmatrix} \quad \frac{\partial \mathbf{u}^*}{\partial p_4} = \begin{bmatrix} 0.019085 \\ 0.002402 \\ 0.159647 \\ 0.083860 \end{bmatrix} \quad \text{varied } p_4 \text{ from } 4.0 \text{ to } 6.0$$

Special features: Hessian matrix is available. Cubic objective function and linear constraints. Active set changes are introduced for changes in p_4 . The parameters being studied are right hand side perturbation. This problem has also been studied by Fiacco et. al. (1974,1983).

A plot of the sensitivity of the optimum with respect to p_1 is presented in figure A.9. Plots of the optimum sensitivity with respect to p_2 and p_3 are similar to those to p_1 . For variation in the first three parameters, the optimum objective function behaves linearly and there are no changes in the active set for the range of parameter perturbation that was studied.

A plot of the sensitivity of the optimum with respect to p_4 is presented in figures A.10 and A.11. Constraint g_9 leaves the active set when $p_4 \leq 4.876$. After constraint g_9 has left the active set further variation of p_4 has no effect on the optimum. For increasing values of p_4 , errors would be introduced if a linear extrapolation was used.

Figure A.9 Sensitivity with respect to $p(1)$ for Problem 4

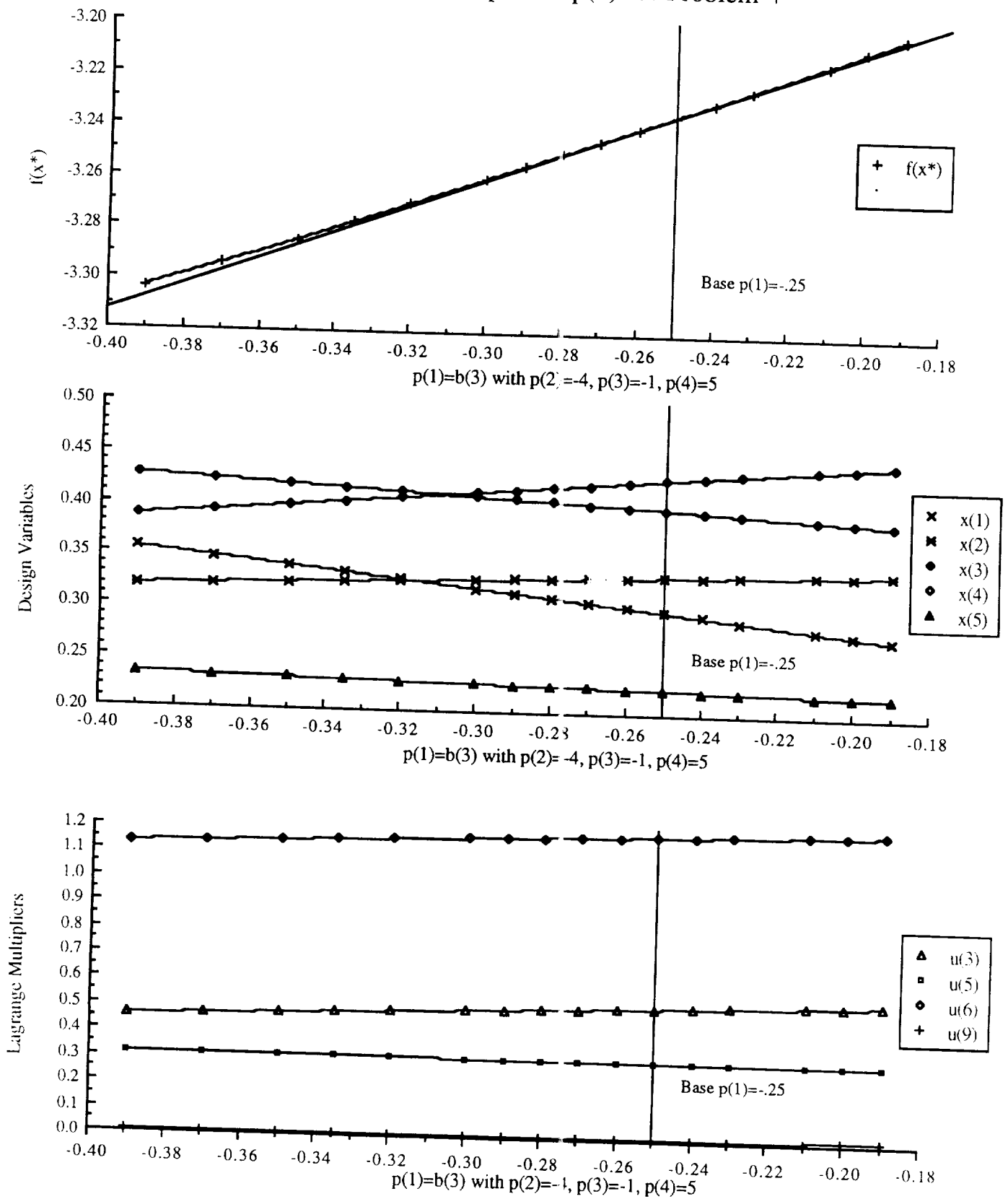


Figure A.10 Sensitivity with respect to $P(4)$ for Problem 4

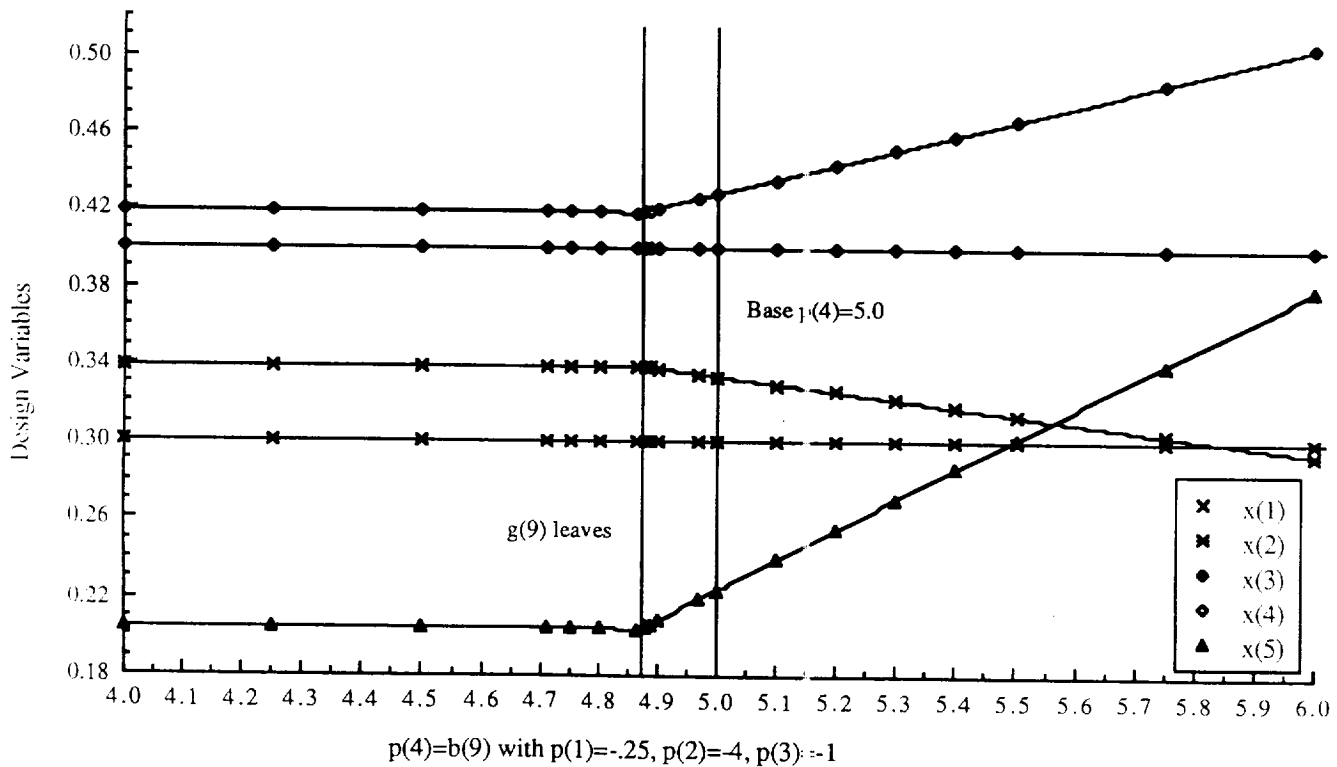
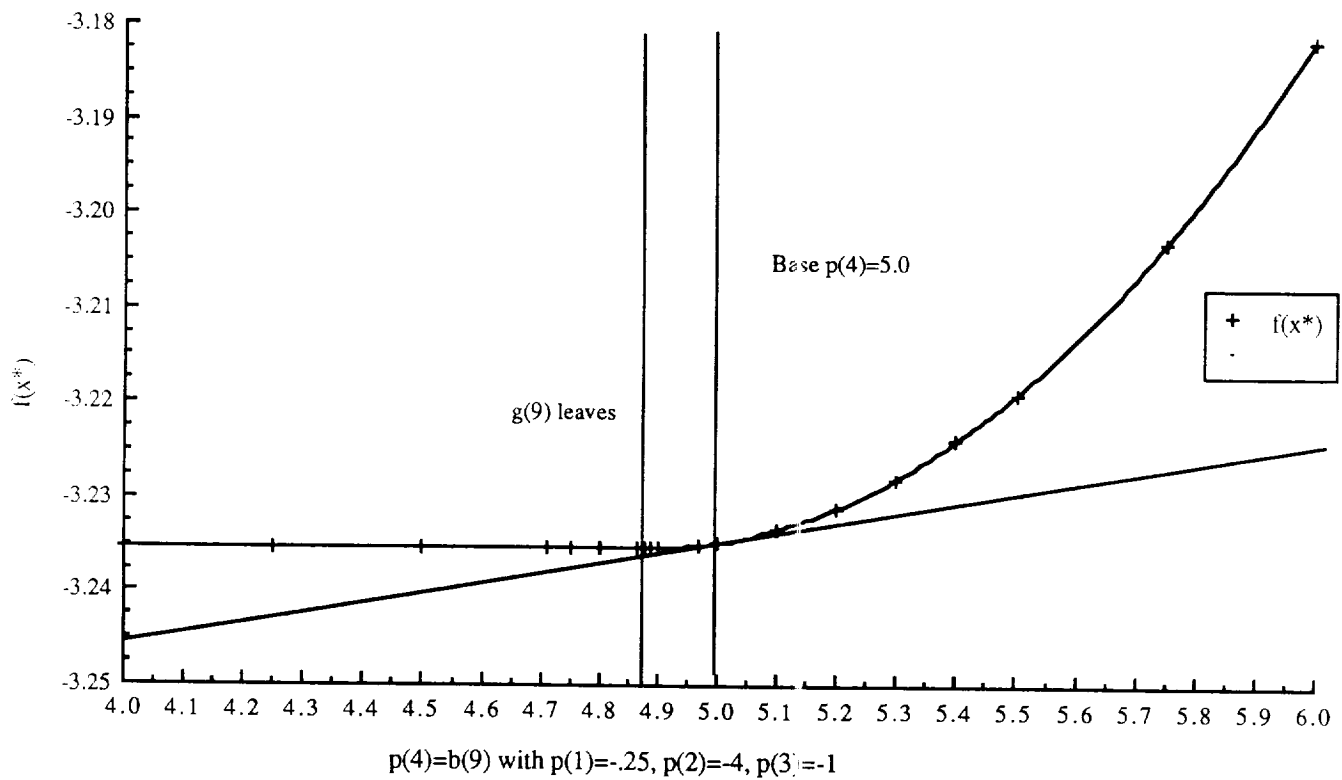
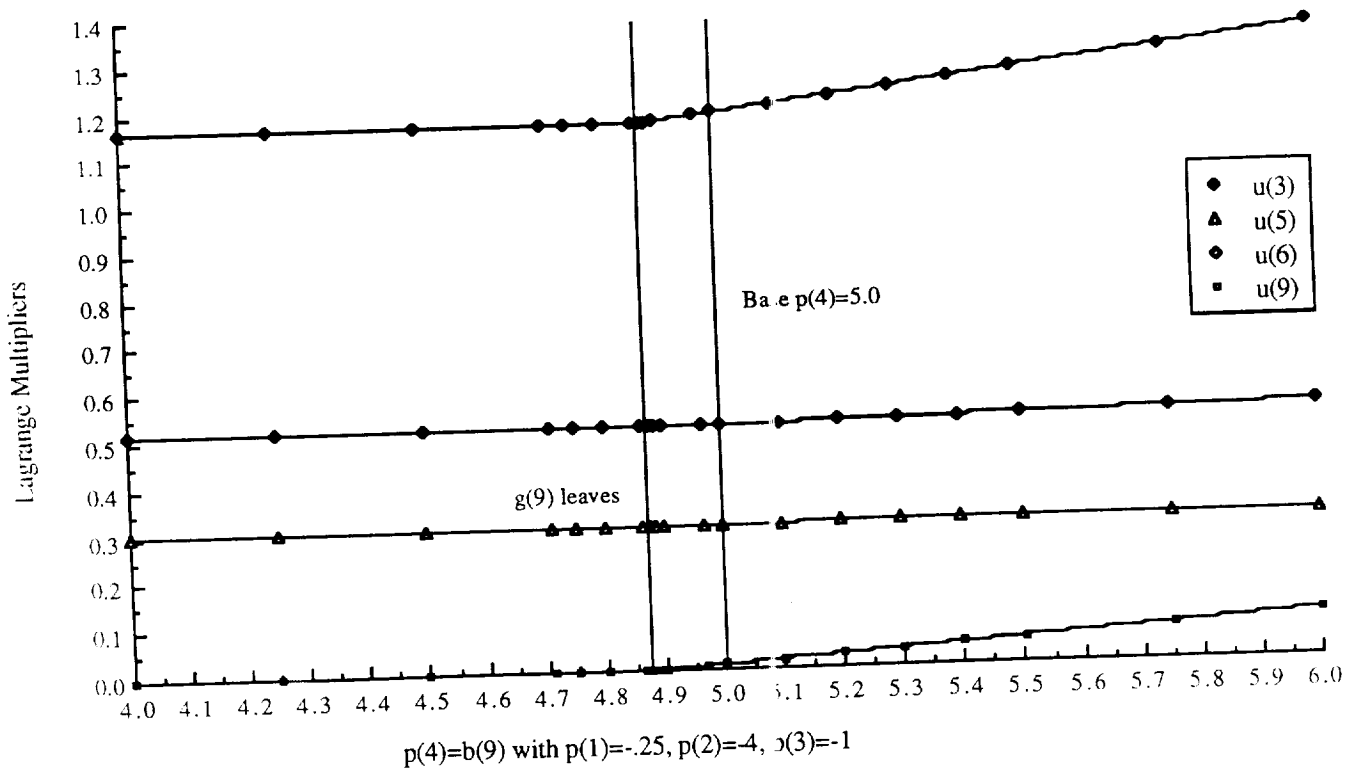
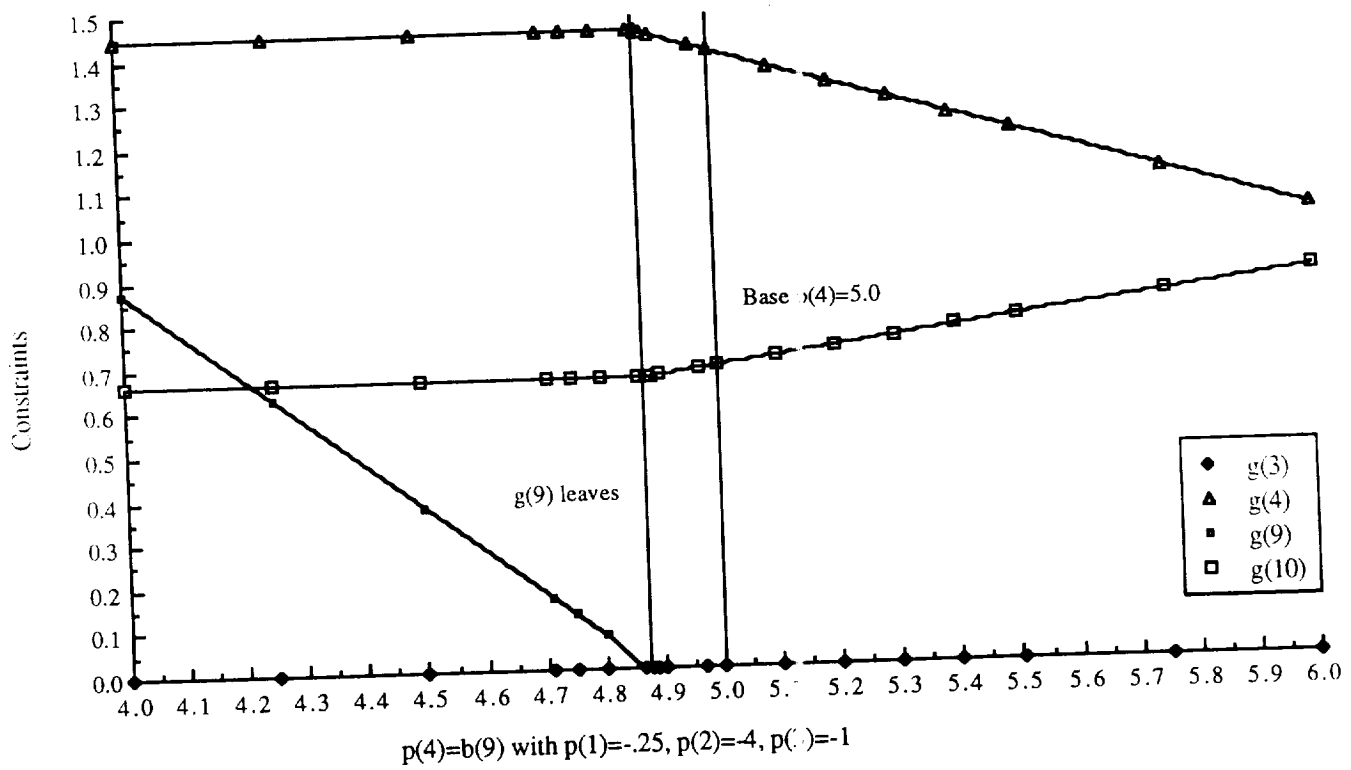


Figure A.11 Sensitivity with respect to $P(4)$ for Problem 4 (cont)



Appendix 2

THE RQCRE AND RQSEN SYSTEMS

This section will discuss the system that was created for studying parameter sensitivities. The first subsection introduces the support program RQCRE that was written to simplify the construction of test problems. The second section will discuss the RQSEN system. The RQSEN system is an interactive program that acts as a post processor/sensitivity analysis module for the RQOPT program.

RQCRE and RQSEN were set up to have some user friendly features. The programs have some user-friendly features however the programs will crash if, the user enters numbers in response to a prompt for an alpha data type, and may crash if the user enters letters when the system is requesting numbers.

A2.1 The RQCRE Support System

The RQCRE program was written to reduce the time required to implement test problems. The RQSEN program requires approximately 30 arrays and a complicated main program to be written by the user. The RQCRE program automatically dimensions the proper arrays and writes the required calling programs. Using the RQCRE program reduced the time required to implement test programs during our initial testing.

The RQCRE program is essentially a program that writes another program. The main features of the RQCRE program are;

1. The program can be used in an interactive mode.
2. The program writes the main calling program.
3. The program can write an outline of the function subprogram.
4. The program can be used to update the problem formulation.

A structure chart of the RQCRE program (in CMS) is presented in Figure A2.1. The basic functions of the program modules are;

RQCRE.EXEC - this module connects the proper output files to the proper unit numbers.

RQCRE - is a FORTRAN program that can be used interactively to create a problem for submission to the RQSEN system. The input to RQCRE can either come from a data file or from the user. The output from RQCRE is a data file "data" that can be used as an input file for RQSEN, a MAIN FORTRAN

program ready for compilation, and a shell for the function subprogram.

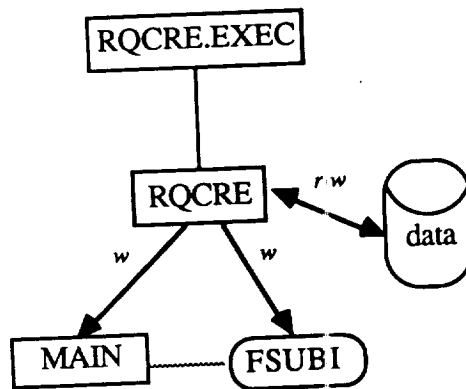


Figure A2.1 A Structure Chart for the RQCRE Program.

```

3      =N
0      =LFSW
0      =NEQ
0      =LEQ
2      =NINEQ
2      =LINEQ
2      =NPARM
15     =MAXIT
1      =ISTART
1      =IFIN
50     =LPP
6      =NOUTFL
5      =ITR
1.E-4   =PMIN
1.E-8   =CAPFMN
0.5     =DELTA
10.0    =R
.0001   =GAMMA
1.0E-8  =EPSQP
5.E-4   =EPSGRD
2       =IDIF
0       =NSCALE
1.E-11  =ZEROM
1.1     =X(1)
1.2     =X(2)
1.3     =X(3)
-10.    =XMIN(1)
-10.    =XMIN(2)
-10.    =XMIN(3)
20.0    =XMAX(1)
20.0    =XMAX(2)
20.0    =XMAX(3)
1.0     =PARM(1)
3.0     =PARM(2)
  
```

Figure A2.2 A Sample of a Data file (Test Problem 2) for the RQSEN Program.

data - the input/output data file that contains the algorithm parameters, starting point and initial values of the parameters for RQSEN.

MAIN - FORTRAN program used as the main calling program when running RQSEN.

FSUBI - a FORTRAN function that the user is required to modify by adding the definitions of the constraints and objective function.

A sample of the data file for test problem 2 is presented in Figure A2.2. This file was written by the RQCRE program. This data file is used as an input to the RQSEN system to provide the programs with the values of the algorithm parameters, design variables, and initial values of the parameters.

A sample of the a program written by the RQCRE program is provided in Figure A2.3. The program represents an implementation for test problem 2 (described in appendix 1). The only modifications that were made to the program are, the objective function and the constraint definitions that were added to the code generated by RQCRE.

A2.2 The RQSEN program

This section describes the implementation of the RQSEN system. The first topic to be discussed is the capabilities of the RQSEN system. Next a description of the implementation is provided. The final topic presented in this section is a sample session from the RQSEN system.

The basic capabilities of the RQSEN system are;

1. The program can be used to solve optimal design problems.
2. The program can be used to conduct convergence studies for various versions of RQOPT.
3. The program can be used to calculate parameter sensitivity derivatives.
4. The program can be used to conduct studies of large variations in the parameters.
5. The program can be used to create parameter sensitivity plots that can be used for trade off studies.

The RQSEN system is currently implemented on the following systems, an IBM 4341 under the CMS operating system and a microVAX under the VMS operating system. The programs are written in FORTRAN 77 and implemented in double precision.

Figure A2.4 presents a structure chart for the RQSEN program (CMS

```

      PROGRAM RTS02
C
C TEST PROBLEM 2 PH. D., CREATED BY TODD J. BELTRACCHI TO EXAMINE
C CONVERGENCE OF THE HESSIAN APPROX AND CHANGES IN THE ACTIVE SET
C THIS HAS A QUADRATIC OBJECTIVE FUNCTION AND LINEAR CONSTRAINTS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(3),XMIN(3),XMAX(3),SCALE(3),H(3,3),DELFHG(3,3),
1  FUNCT(3),P(3),U(8),V(1),XS(3),DHFHS(3,3),FUNCTS(3),PS(3),US(8),
2  VS(1),DFDP(2),FUNCTP(3,2),DXDP(3,2),DUDP(8,2),DVDP(1,2),DFDPE(2),
3  DGDP(2,2)
      LOGICAL YESNO,ACTPT(8),ACTPTS(8)
      CHARACTER*3 YSN
      CHARACTER*7 FILENM
      COMMON /OPTDAT/ D(400)
      COMMON /BFSWK/ DD(20)
      COMMON /PMINI/ PMINI(3)
      COMMON /PMAXI/ PMAXI(3)
      COMMON /PARMS/ PARM(2)
      INCLUDE (RQS)
      END
C *****
      FUNCTION FSUBI(X,IEVAL)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(3)
      COMMON /NFEVAL/ NCE,NFE
      COMMON /PARMS/ P(2)
      IF (IEVAL .GT. 1) GOTO 2
1    NFE=NFE+1
C PLACE OBJECTIVE FUNCTION DEFINITION HERE
      FSUBI=2.5*(X(1)**2+X(2)**2+X(3)**2)+X(1)*X(2)+X(1)*X(3)+X(2)*X(3)
1    +5.*(X(1)+P(1)*X(2)+X(3))
      RETURN
2    NCE=NCE+1
C PLACE LINEAR EQUALITY CONSTRAINTS HERE
      GOTO (10, 20, ) IEVAL
10    FSUBI=P(1)*X(1)+X(2)+X(3)-P(2)
      RETURN
20    FSUBI=X(1)+2.*X(2)+3.*X(3)-4.7-E(1)
      RETURN
      END

```

Figure A2.3 A Sample Program (Test Problem 2) for RQSEN.

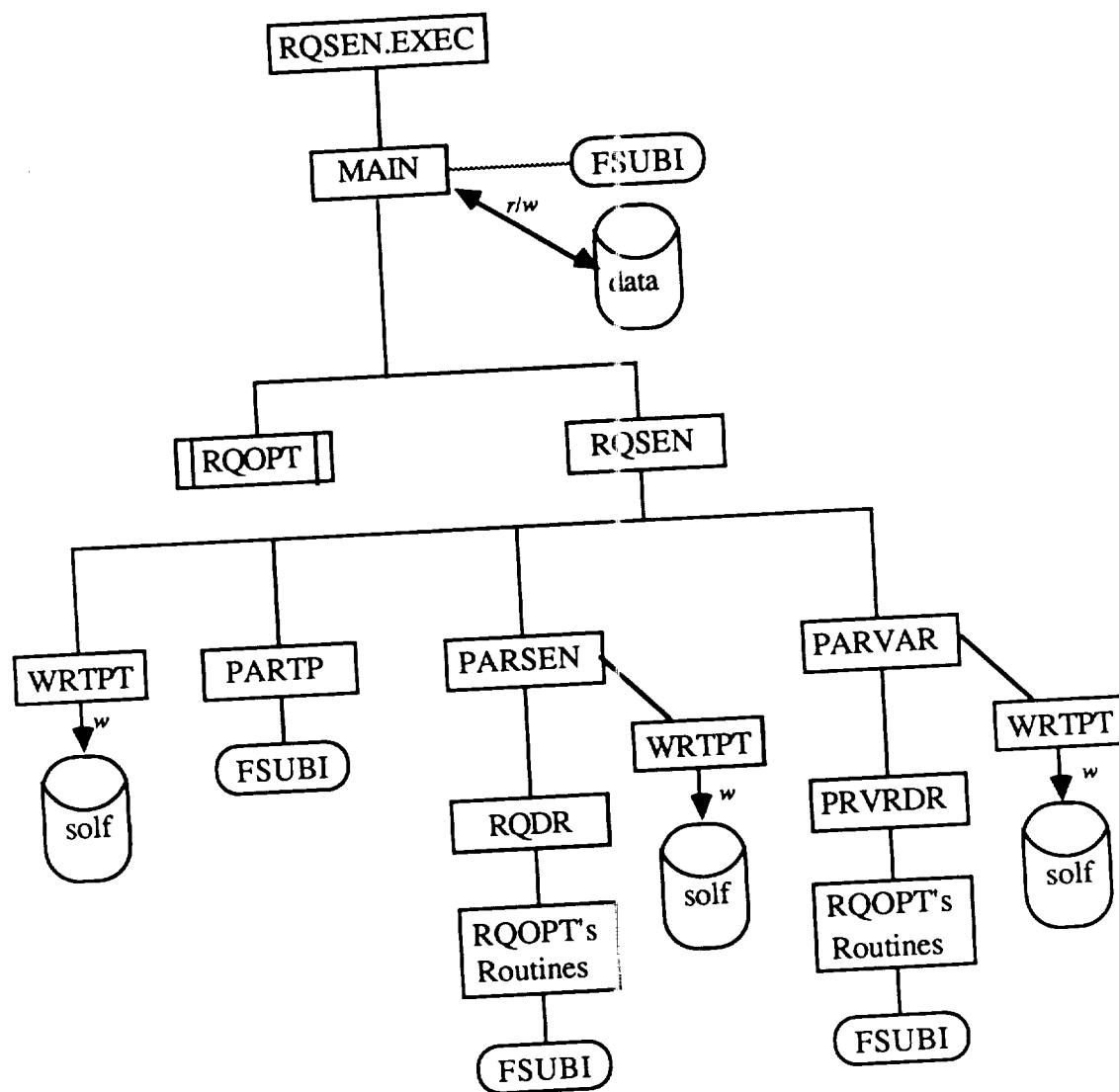


Figure A2.4 A Structure Chart for the RQSEN System.

implementation). A brief explanation of each of the program modules is provided

RQSEN.EXEC - this module connects the proper files to the proper unit numbers and prompts the user for the name of the program to be run.

MAIN - the main calling program. This module calls RQOPT and RQSEN, this module also reads in the starting point and algorithm parameters for RQOPT and allows the user to save solution point to a data file for later use, i.e. a sensitivity study at a later time.

FSUBI - the function subprogram that defines the objective function and constraints.

data - the data file that contains the algorithm parameters and values of the design variables.

RQOPT - implementation of the RQP method described in Beltracchi and Gabriele (1987 b).

RQSEN - the main driving routine for a sensitivity analysis.

WRTPT - A utility program for writing the design point, Lagrange multipliers and values of the objective function and constraints to a summary file. The summary file is in a form which can be read by a plotting program to graphically display the sensitivity information.

self - a data file used to communicate optimum design points to a plotting program.

PARTP - a subroutine that uses either forward, central or user supplied routines to calculate the partials with respect to the design parameters of the objective function and constraints.

PARSEN - a subroutine that calculates $\partial x^*/\partial p$, $\partial u/\partial p$, df/dp and dg/dp by either forward or central differencing. The user can specify the perturbation size and the number of iterations that RQOPT uses to solve the perturbed problem.

PARSEN2 - a subroutine similar to PARSEN but this implements modified central differencing.

RQDR - is the routine that controls the execution of RQOPT for reoptimization.

PARVAR - the subroutine used to conduct studies of large variations in problem parameters, if parameter sensitivity derivatives are available then the location of the starting point for the reoptimization is approximated by.

$$x_{\text{initial}} = x(p) + \frac{\partial x}{\partial p} \Delta p$$

PRVRDR - is the routine that controls the execution of RQOPT for reoptimization.

The rest of this section describes the steps involved in using the RQSEN system.

The first step is for the user to create the necessary FORTRAN code to define the main calling program and the function subprogram (similar to the one in Figure A2.3). The second step is to set up a data file (similar to the one presented in Figure A2.2) with the values of the design variables and algorithm parameters. These first two steps can be performed with the aid of the RQCRE preprocessor.

Once the calling program, function subprogram, and the data file are defined, the user can run the RQSEN system to conduct a study of the sensitivity of the optimum of the problem or to study the convergence of the problem.

A sample of some convergence plots are presented in Figure A2.5 & 6. These plots can be used to assess the convergence criteria of various algorithms, for example Figure A2.5 shows that the BFS version was able to solve the test problem faster than the version that used the SR1 update. However the SR1 update found a more accurate estimate of the optimum, and once the region of the minimum was located the convergence for the SR1 update was better than the BFS update.

The best way to explain how the program can be used to conduct parameter sensitivity studies is to provide a sample session (see Figure A2.7) that was run under the IBM version of RQSEN. The next several paragraphs describe the output in Figure A2.7, all user responses are shown in *italics*.

The sample problem 2, described in appendix 1, is solved. The modified SR1 update is used to approximate the Hessian of the Lagrangian, and modified central differencing is used to calculate the sensitivity derivatives. The first step for running RQSEN to invoke the exec file to start the program, this is done by entering *rqsr14*. The first prompt asks the user for a name of a data file to store the results in the user responds with *rts02*. This data file can be used to maintain a summary of all optimum points. The next prompt is for the name of the program to be run, in our example program *rts02*, an implementation of test problem 2 (see appendix 1) was run. The program and data file were presented in Figure A2.3 and A2.2.

When the program begins the first prompt is for the name of the file with the input data. A response of *rts02* is entered, the program then reads in the data from file *rts02*. The next prompt asks if there is an approximation to several arrays available. These arrays

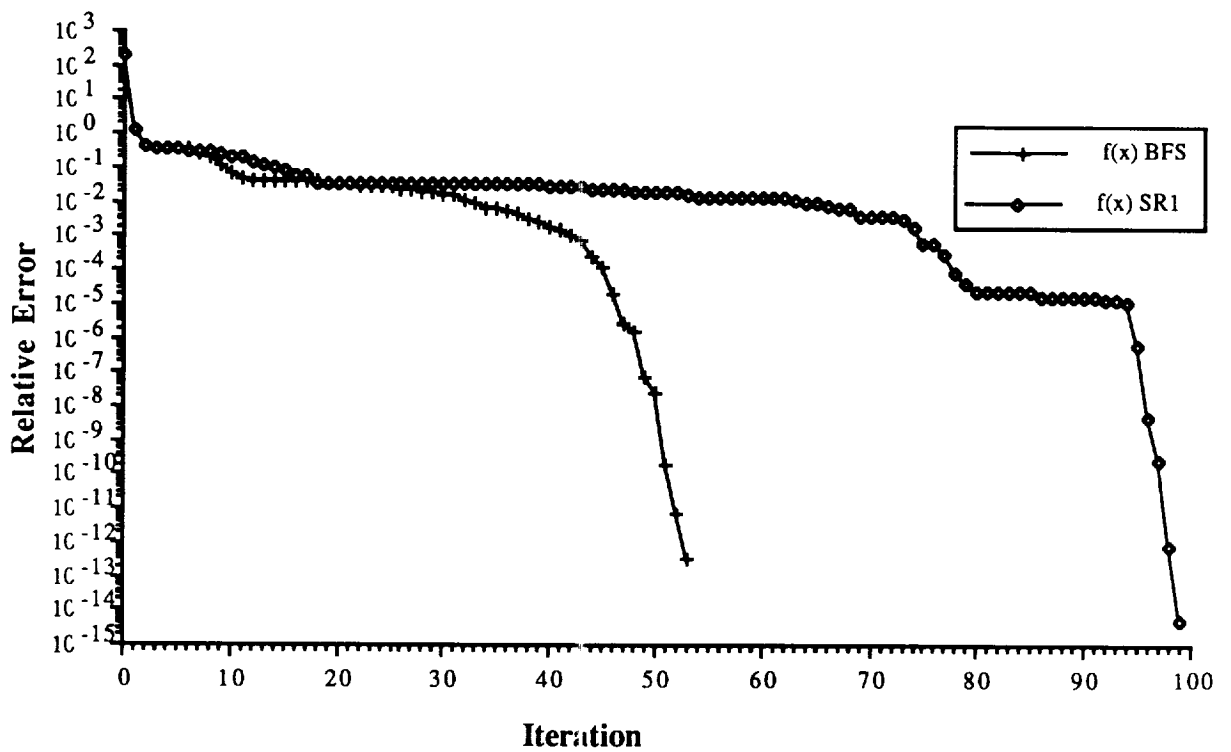


Figure A2.5 A Sample Plot comparing the Convergence Characteristics of the BFS and SR1 Updates.

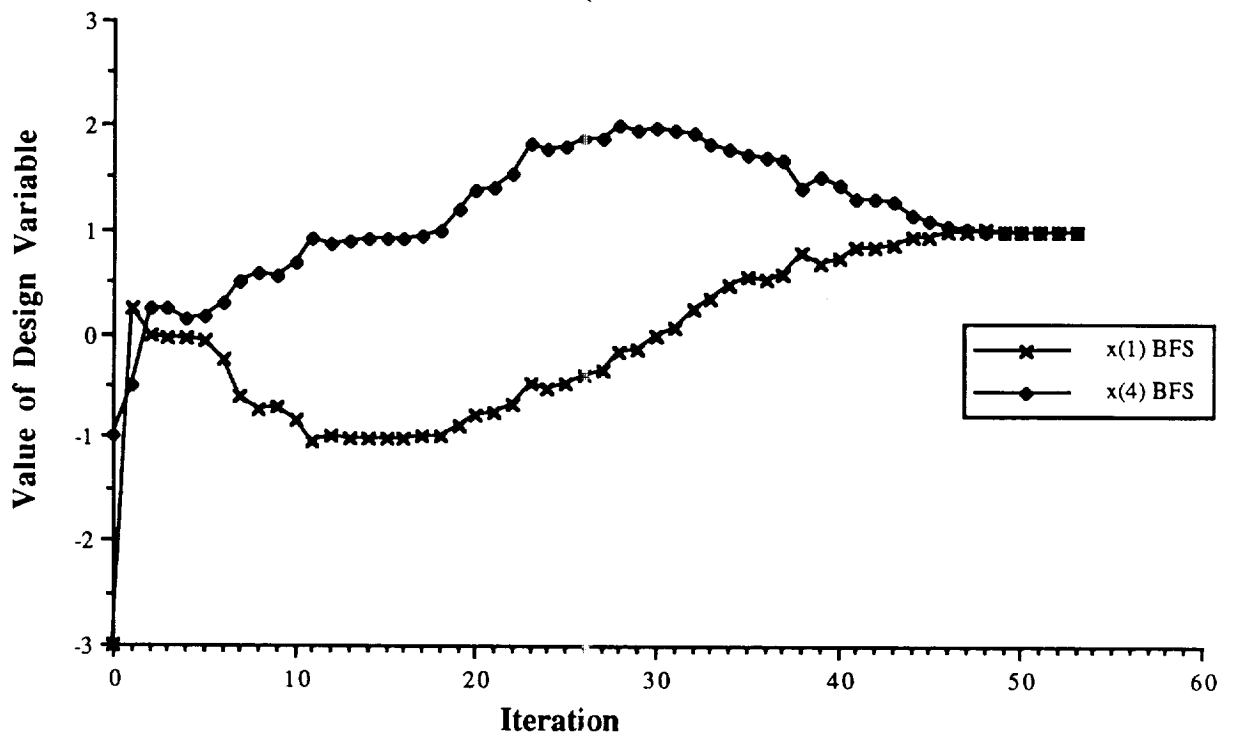


Figure A2.6 A Sample Plot Showing the Convergence of the Design Variables.

are saved if the optimum has already been found, a response of "n" for no is entered¹. Now the program invokes the RQOPT program to locate the optimum of the problem. A summary of the output from RQOPT (during the optimization) is presented in Figure A2.7(a-c). After the problem has been solved some final statistics from RQOPT are presented along with an approximation to the Hessian of the Lagrangian, in both the LDL^T format and the unfactored form. These are shown in Figure A2.7(b & c).

After the problem has been solved by RQOPT control is returned to the preprocessor (see Figure A2.7(c)). The preprocessor provides the user with a choice of being able to save the optimal point. In the example the response was "y" for yes was entered, next the user is asked if he wants to save the final point in the same file as the initial point, a response of "n" for no was entered². Next the user is asked to supply a new name of the data file to store the point in, a response of *rts02s* was entered. The data file was then written and the user asked if they wanted the gradients and Hessian approximation to be written to the file, a response of "y" was entered³. The preprocessor next asks if the user wants to perform a sensitivity analysis, a response of "y" for yes is entered.

Now control is passed to the RQSEN program (see Figure A2.7(c-e)). The first question asked by RQSEN is if the user wants the solution points written to a solution file a response of "n" is entered. The next question asked is for a value of epsilon to calculate the partial derivatives of the problem functions. RQSEN then calculates the partials of the objective function and constraints, the derivative of the objective function is then calculated by equation 1.20.

The next step in the sample output is the calculation of the partials of the optimum design variables and Lagrange multipliers with respect to the first parameter for the problem. Again the user can specify the size of the perturbation of the parameter and the number of iteration that RQOPT is allowed to use for solving the perturbed problem. In this example central differencing is used, and Hessian updating is allowed, notice that in Figure A2.7(d) the Hessian approximation has converged.

Figure A2.7(e) shows the values of the parameter sensitivity derivatives that were calculated by RQSEN. The gradient of the objective function df^*/dp was calculated by 3

¹RQCRE or RQSEN will accept either "YES", "YE", "Y", "yes", "ye", or "y" for a yes response and "NO", "N", "no", "n", null for a no response.

²If the user responded yes then the final point would overwrite the initial point in the data file.

³ The Data file can now be used as an input to RQSEN for a sensitivity analysis performed using the gradients and Hessian approximation that were found when solving the problem.

different approximations, all values are reported for comparison. The next option of the program is to study finite perturbations in the parameter. This option can be used to calculate optimal designs for different values of the parameters and then to write the optimal design points to a data file (see Figure A2.7(e&f)). The rest of the sample output involves terminating the program.

When the user saves the optimal points plots of the optimum sensitivity can be made. An example of the sensitivity for problem 2 when p_1 is varied is presented in Figure A2.8.

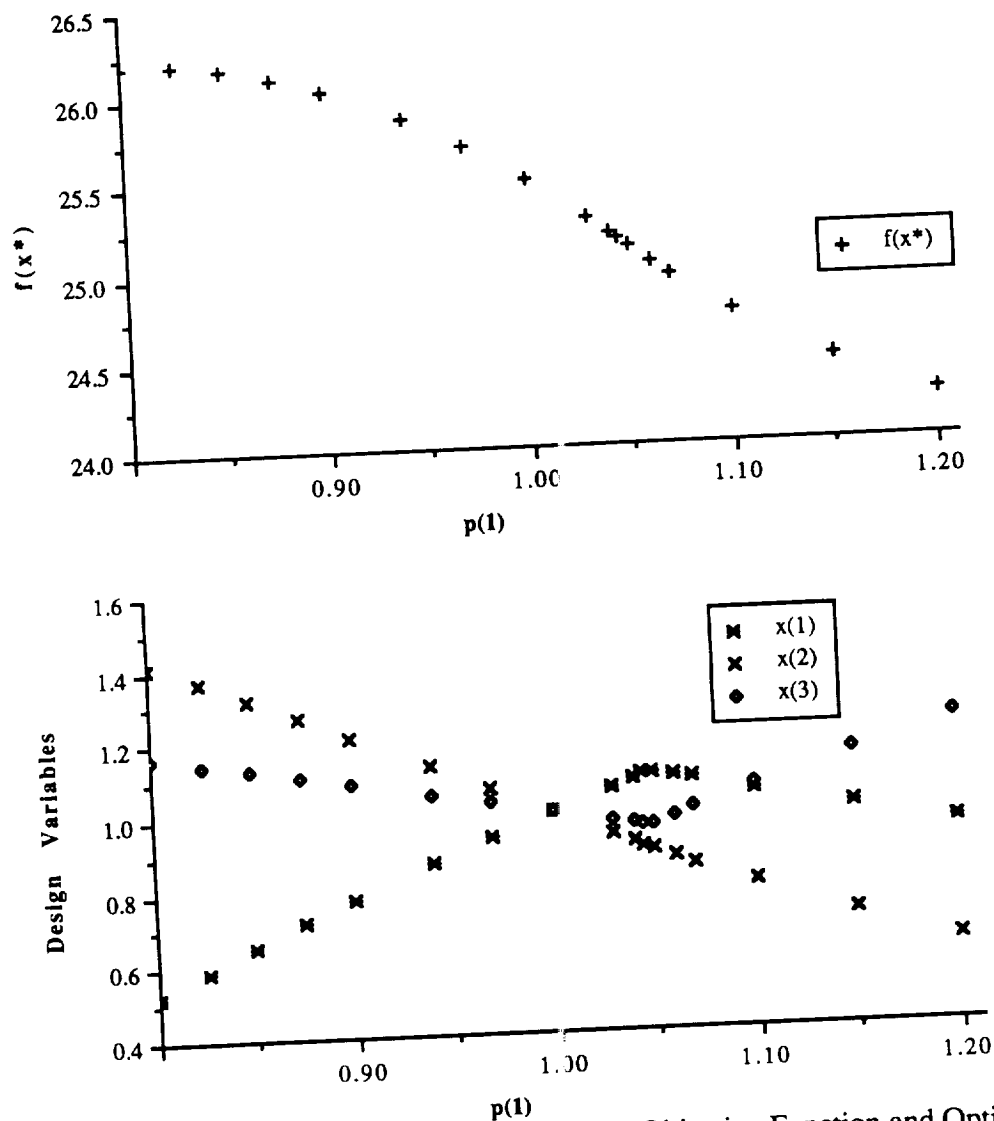


Figure 3.8 A Plot of the Sensitivity of the Optimum Objective Function and Optimum value of the Design Variables for Problem 2 when p_1 is perturbed

```

rqsrl4
  INPUT NAME OF THE FILE TO HOLD THE SENSITIVITY DATA
rts02
FILEDEF 9 DISK RTS02 DATA A1 ( LRECL 800
  INPUT NAME OF THE PROGRAM TO BE RUN BY RQSEN
rts02
LOAD RTS02 SR14 RQOPT OPTQP RQSEN PARSEN2 PARVAR WRTPT ( CLEAR START
Execution begins...
*****
*      ENTERING RQOPT/RQSEN PREPROCESSOR      *
*****

INPUT NAME OF THE FILE WITH INPUT DATA(XXXXXXX)
rts02
OPENING FILE, RTS02  ON UNIT= 3 FOR INPUT OF PROBLEM DATA
  ARE FUNCT,H,U,V,ACTPT,DELPHG IN FILE RTS02  (YES/NO)
n
RQOPT VERSION 2.12 - EXPERIMENTAL 2/9/88

STARTING INFORMATION

INPUT PARAMETERS:
      NUMBER OF VARIABLES = 3
OPTIMIZATION TO BE PERFORMED ON A NON-LINEAR OBJECTIVE FUNCTION
      NUMBER OF EQUALITY CONSTRAINTS = 0
      NUMBER OF LINEAR EQ CONSTRAINTS = 0
      NUMBER OF INEQUALITY CONSTRAINTS = 2
      NUMBER OF LINEAR INEQ CONSTRAINTS = 2
      MAXIMUM NUMBER OF ITERATIONS = 100
      LINES PER PAGE = 50
      JTR = 1
      NUMBER OF OUTPUT FILE = 6

      DELTA = 0.500000E+00
      R = 0.100000E+02
      GAMMA = 0.100000E-04
      EPSILON FOR THE QP = 0.100000E-09
      EPSILON FOR THE GRAD = 0.100000E-03
      DIFFERNING TYPE = 2
      INITIAL VAULE OF CAPF = 0.000000E+00
      MINIMUM VALUE OF CAPF = 0.100000E-06
      MINIMUM NORM OF P VECTOR = 0.100000E-03
      SCALING PARAMETER NSCALE = 0
      NUMBER OF PARAMETERS NPARM = 2
INDEX  PARAMETER VALUE
      1  1.0000000000000000
      2  3.0000000000000000

      I      XMIN      XMAX      SCALE
      1  0.000000E+00  0.100000E+03  0.100000E+01
      2  0.000000E+00  0.100000E+03  0.100000E+01
      3  0.000000E+00  0.100000E+03  0.100000E+01

```

Figure A2.7(a) A Sample of RQSEN for Test Problem 2.

THE HESSIAN APPROXIMATION IN LDL(T) FORM
 ROW 1 0.100000E+01

ROW 2 0.000000E+00 0.100000E+01

ROW 3 0.000000E+00 0.000000E+00 0.100000E+01

ENTERING DRIVER ROUTINE
 -----ITERATION 0: 0-----
 PAGE 1 OBJECTIVE FUNCTION = 0.33160000E+02 FUNCTION EVALUATIONS= 1
 CONSTRAINT EVALUATIONS= 14
 INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
 1 0.1100000E+01 0.600000E+00 0.000000E+00
 2 0.1200000E+01 0.170000E+01 0.000000E+00
 3 0.1300000E+01
 CHECKING CONVERGENCE THE NORM OF P= 1.96446237582566496
 -----ITERATION 1: 0-----
 PAGE 1 OBJECTIVE FUNCTION = 0.25535356E+02 FUNCTION EVALUATIONS= 8
 CONSTRAINT EVALUATIONS= 19
 INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
 1 0.9059701E+00 A-.288658E-14 0.000000E+00
 2 0.1000000E+01 A0.488060E+00 0.000000E+00
 3 0.1094030E+01
 CHECKING CONVERGENCE THE NORM OF P= 0.345110459418981358
 -----ITERATION 2: 1-----
 PAGE 1 OBJECTIVE FUNCTION = 0.25503133E+02 FUNCTION EVALUATIONS= 16
 CONSTRAINT EVALUATIONS= 23
 INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
 1 0.1027985E+01 A0.577316E-14 0.586085E+01
 2 0.1000000E+01 A0.244030E+00 0.659360E-01
 3 0.9720149E+00
 CHECKING CONVERGENCE THE NORM OF P= 0.395769395140752175E-01
 -----ITERATION 3: 2-----
 PAGE 1 OBJECTIVE FUNCTION = 0.25500000E+02 FUNCTION EVALUATIONS= 23
 CONSTRAINT EVALUATIONS= 25
 INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
 1 0.1000000E+01 A0.288658E-14 0.120000E+02
 2 0.1000000E+01 A0.300000E+00 -.288658E-14
 3 0.1000000E+01
 CHECKING CONVERGENCE THE NORM OF P= 0.395476598531791093E-08
 CONVERGENCE ACHIEVED

FINAL STATISTICS

CONVERGENCE ACHIEVED

IN 29 FUNCTION EVALUATIONS
 4 FUNCTION GRADIENTS
 25 CONSTRAINT EVALUATIONS
 2 CONSTRAINT GRADIENTS
 3 ITERATIONS

WITH 0.00000000E+00 BEING THE MAXIMUM CONSTRAINT VIOLATION

Figure A2.7(b) A sample of RQSEN for test problem 2.

THE HESSIAN APPROXIMATION IN LDL(T) FORM

ROW 1 0.450000E+01

ROW 2 0.444444E+00 0.211111E+01

ROW 3 0.111111E+00 0.842105E+00 0.294737E+01

THE HESSIAN APPROXIMATION UPPER TRIANGLE

ROW 1 0.450000E+01 0.200000E+01 0.500000E+00

ROW 2 0.300000E+01 0.200000E+01

ROW 3 0.450000E+01

PAGE 1 OBJECTIVE FUNCTION = 0.2550000E+02 FUNCTION EVALUATIONS= 29

CONSTRAINT EVALUATIONS= 25

INDEX	X(I)	H(I) = ?0	V(I)	G(I) >= ?0	U(I)
1	0.1000000E+01			A0.288658E-14	0.120000E+02
2	0.1000000E+01			A0.300000E+00	0.000000E+00
3	0.1000000E+01				

DO YOU WISH TO SAVE THE FINAL DATA (YES/NO)?

y

DO YOU WISH TO USE THE SAME FILE RTS02 (YES/NO)?

n

INPUT NAME OF THE FILE FOR STORAGE OF DATA (XXXXXXX)

rts02s

OPENING FILE, RTS02S ON UNIT= 4 TO STORE PROBLEM DATA

DO YOU WANT FUNCT, H, U, V, DELFHG, ACTPT WRITTEN TO RTS02S (YES/NO)?

y

DO YOU WANT TO PERFORM A SENSITIVITY ANALYSIS (YES/NO)?

y

WELCOME TO RQSEN1.0

A SENSITIVITY ANALYSIS PROGRAM FOR RQOPT

LAST MODIFIED APRIL 28 1988

DO YOU WANT TO WRITE THE SOLUTION POINTS TO FILE= 9 (YES/NO)?

n

THE DERIVATIVE OF THE OBJECTIVE FUNCTION WITH
RESPECT TO ALL PARAMETERS WILL BE CALCULATED

INPUT EPSP FOR THE CALCULATION OF DF/DP

?

.0001

THE DERIVATIVE OF OBJECTIVE FUNCTION W.R.T. P

-0.6999999989E+01 0.1200000000E+02

DO YOU WANT TO STUDY FINITE PERTURBATIONS

ENTER PARAMETER NUMBER OR (-1 OR CTRL Z) TO CALCULATE GRADS?

?

-1

DO YOU WISH TO FIND PARTIALS OF THE DESIGN VARIABLES AND

LAGRANGE MULTIPLIERS W.R.T. PARAMETER (NUMBER OR -1 TO END)?

?

1

Figure A2.7(c) A Sample of RQSEN for Test Problem 2.

```

ENTER EPSP FOR THE GRADIENT CALCULATION?
?
.0001

PERFORMING A SENSITIVITY ANALYSIS FOR PARM( 1)
ASSUMING BASE POINT IS STABLE

BASE POINT VALUE PARM( 1)=0.1000000000000000E+01
PERTURBED VALUE OF PARM( 1)=0.1000000000000000E+01

ENTER THE NUMBER OF ITERATIONS FOR RQOPT?
?
2
***** ENTERING RQOPT *****

ENTERING DRIVER ROUTINE
-----ITERATION 0: 0 PARAMETER( 1)= 0.10001000E+01-----
PAGE 1 OBJECTIVE FUNCTION = 0.25500500E+02 FUNCTION EVALUATIONS= 7
CONSTRANT EVALUATIONS= 26
INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
1 0.1000000E+01 A0.100000E-03 0.120000E+02
2 0.1000000E+01 A0.299900E+00 0.000000E+00
3 0.1000000E+01
CHECKING CONVERGENCE THE NORM OF P= 0.924648797898831832E-03
-----ITERATION 1: 0 PARAMETER( 1)= 0.10001000E+01-----
PAGE 1 OBJECTIVE FUNCTION = 0.25499301E+02 FUNCTION EVALUATIONS= 14
CONSTRANT EVALUATIONS= 28
INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
1 0.1000483E+01 A-.133227E-14 0.119995E+02
2 0.9992333E+00 A0.299400E+00 0.000000E+00
3 0.1000183E+01
CHECKING CONVERGENCE THE NORM OF P= 0.673599522571675425E-03
-----ITERATION 2: 0 PARAMETER( 1)= 0.10001000E+01-----
PAGE 1 OBJECTIVE FUNCTION = 0.25499300E+02 FUNCTION EVALUATIONS= 21
CONSTRANT EVALUATIONS= 30
INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
1 0.1000208E+01 A-.222045E-14 0.119995E+02
2 0.9997833E+00 A0.299400E+00 0.000000E+00
3 0.9999083E+00
***** LEAVING RQOPT *****
THE HESSIAN APPROXIMATION UPPER TRIANGLE
ROW 1 0.500000E+01 0.100000E+01 0.100000E+01
ROW 2 0.500000E+01 0.100000E+01
ROW 3 0.500000E+01

PERTURBED VALUE OF PARM( 1 )= 0.9998999999999999997

ENTER THE NUMBER OF ITERATIONS FOR RQOPT?
?
1

```

Figure A2.7(d) A Sample of RQSEN for Test Problem 2.


```

THE NEW VALUE OF P( 1)= 0.11000000E+01
***** ENTERING RQOPT *****

ENTERING DRIVER ROUTINE
-----ITERATION 0: 0 PARAMETER( 1)= 0.11000000E+01-----
PAGE 1 OBJECTIVE FUNCTION = 0.24893909E+02 FUNCTION EVALUATIONS= 30
CONSTRANT EVALUATIONS= 60
INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
1 0.1208319E+01 A0.208111E-01 0.115333E+02
2 0.7833292E+00 A-.300030E+00 0.000000E+00
3 0.9083308E+00
CHECKING CONVERGENCE THE NORM OF P= 0.219176271755135058
-----ITERATION 1: 0 PARAMETER( 1)= 0.11000000E+01-----
PAGE 1 OBJECTIVE FUNCTION = 0.24742404E+02 FUNCTION EVALUATIONS= 37
CONSTRANT EVALUATIONS= 62
INDEX X(I) H(I) = ?0 V(I) G(I) >= ?0 U(I)
1 0.1045910E+01 A0.415223E-13 0.104876E+02
2 0.7944067E+00 A0.235367E-13 0.542741E+00
3 0.1055092E+01
CHECKING CONVERGENCE THE NORM OF P= 0.000000000000000000E+00
CONVERGENCE ACHIEVED
***** LEAVING RQOPT *****
ENTER THE PERTURBATION FOR THE PARAMETER OR
ENTER 0.0 OR NULL TO EXIT FORM THIS SUBROUTINE?
?
0.0
DO YOU WANT TO STUDY FINITE PERTURBATIONS
ENTER PARAMETER NUMBER OR (-1 OR CTRL Z) TO CALCULATE GRADS?
?
-1
DO YOU WISH TO FIND PARTIALS OF THE DESIGN VARIABLES AND
LAGRANGE MULTIPLIERS W.R.T. PARAMETER (NUMBER OR -1 TO END)?
?
-1
Ready; T=2.81/4.12 11:23:12 $2.46

```

Figure A2.7(f) A Sample of RQSEN for Test Problem 2.

REFERENCES

- Adelman, H. M. and Haftka, R. T., 1986, "Sensitivity Analysis of Discrete Structural Systems", in *AIAA Journal*, Vol 24, No. 5, May 1986, pp 823-832.
- Allen, R. G. D., 1938, *Mathematical Analysis for Economists*, Macmillan, New York.
- Arbuckle, P. D., and Sliwa, S. M., 1984, "Experiences Performing Conceptual Design Optimization of Transport Aircraft" in *Recent Experiences in Multidisciplinary Analysis and Optimization, Part 1*, Compiled by J. Sobieski, NASA CP 2327, pp 87-101.
- Armacost, R. L., and Fiacco, A. V., 1974, "Computational Experience in Sensitivity Analysis for Nonlinear Programming", in *Mathematical Programming*, Vol. 6, pp 301-326.
- Arora, J. S., and Haug, E. J., 1979, "Methods of Design Sensitivity Analysis in Structural Optimization", in *AIAA Journal*, Vol 17, No. 9, pp 970-974
- Arora, J. S., and Tseng, C. H., 1987, "Discussion on ASME Paper No. 86-DET-26," in *ASME Journal of Mechanisms, Transmissions and Automation in Design*, June, Vol 109 No. 2.
- Barthelemy, J. F., and Sobieszczanski-Sobieski, J., 1983, "Extrapolation of Optimum Designs Based of Sensitivity Derivatives," in *AIAA Journal*, Vol. 21, May , pp 797-799.
- Barthelemy, J. F., and Sobieszczanski-Sobieski, J., 1983, "Optimum Sensitivity Derivatives of Objective Functions in Nonlinear Programming" in *AIAA Journal*, Vol. 21, June pp 913-915.
- Bartholomew-Biggs, M. C. 1982," ", in *Nonlinear Optimization 1981*, Edited by M. J. D. Powell, Academic Press, London 1982, pp 213-221
- Bartholomew-Biggs, M. C., 1985, "The Development of Recursive Quadratic Programming Methods Based on the Augmented Lagrangian" Technical Report No. 160 The Hatfield Polytechnic, June.
- Bartholomew-Biggs, M. C., 1986, "Numerical Examples of the Behavior of REQP on Nonlinear Programming Problems Involving Linear Dependence among the Constraint Normals", in *Journal of Optimization Theory and Applications*, vol 48, No 2.
- Bartholomew-Biggs, M. C., 1987, "Recursive Quadratic Programming Methods Based on the Augmented Lagrangian," in *Mathematical Programming Study 31*, Computational Mathematical Programming, edited by K. L. Hoffman, R. H. F. Jackson, and J. Telgen, North Holland - Amsterdam, pp 21-41.
- Bazaraa, M. S., and Shetty, C. M., 1979, *Nonlinear Programming: Theory and Algorithms*, Wiley, New York.
- Belegundu, A., and J. S. Arora, 1985, "A comparison of Structural Optimization," in *International Journal of Numerical Methods in Engineering*.
- Beltracchi, T. J., 1985, *An Investigation of Pshenichnyi's Recursive Quadratic Programming Technique for Engineering Optimization*, MS Thesis Rensselaer Polytechnic Institute, Dec. 1985

Beltracchi, T. J., and Gabriele, G. A., 1987 a, "An Investigation of Pshenichnyi's Recursive Quadratic Programming Technique for Engineering Optimization", in ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol 109 No. 2 pp 248-256 June 1987.

Beltracchi, T. J. and Gabriele, G. A., 1987 b, "RQOPT A FORTRAN 77 Implementation of the Recursive Quadratic Programming Technique for Engineering Optimization, Users Manual, Rensselaer Polytechnic Institute, Department of Mechanical Engineering.

Brayton, R. K. and Cullum, J., "An Algorithm for Minimizing a Differentiable Function Subject to Box Constraints and Errors," in Journal of Optimization Theory and Applications, Vol 29, No 4., pp 521-558 December

Broyden, G. G., 1970, "The Convergence of a class of Double-Rank Minimization Algorithms," in Journal of the Institute of Mathematic Applications, Vol 6., pp 76-90, 222-231.

Buys, J. D. and Gonin, R., 1977, "The use of Augmented Lagrangian Functions for Sensitivity Analysis in Nonlinear Programming." in Mathematical Programming, Vol. 12 No. 2, pp 281-284.

Cha, J. Z., and Mayne R. W. 1987, "Optimization with Discrete Variables Via Recursive Quadratic Programming: Part II: Algorithm and Results," in ASME DE Vol. 10-1, The Proceedings of the 1987 Design Automation Conference, Boston Ma.

Chen, C., Kong, W. C., and Cha, J. Z., 1987, "An Equality Constrained RQP Algorithm Based on the Augmented Lagrangian Penalty Function," ASME paper DAC-87-107.

Cornet B., and Laroque G., 1987, "Lipschitz Properties of Solutions in Mathematical Programming," in Journal of Optimization Theory and Applications, June 1987, pp 407-427.

Coville, A. R., 1968, "A Comparison Study of Nonlinear Programming Codes," Tech Report No. 320-2949, IBM New York Scientific Center, June.

Cullum, J. and Brayton, R. K., 1979 "Some Remarks on the Symmetric Rank-One Update," in Journal of Optimization Theory and Applications, Vol 29, No 4., pp 493-519 December

Dantzig, G. B., 1963, Linear Programming and Extensions, Princeton University Press, Princeton New Jersey.

Dennis, J. E., and Schnable, R. B., 1983, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall.

Diewart, W. E., 1984, "Sensitivity Analysis in Economics", in Computers in Operations Research, Vol. 11, No. 2, pp 141-156.

Dinkel, J. J. and Kochenberger, G. A., and Wong D. S., 1983, "Parametric Analysis in Geometric Programming: An Incremental Approach" in Mathematical Programming with Data Perturbations I, A. V. Fiacco Editor, Lecture notes in Pure and Applied Mathematics, vol 73, Marcel Dekker, New York, pp 93-109.

Dixon, L. C. W., and Szego, G. P., 1980, "The Numerical Optimization of Dynamic Systems: A Survey", In Numerical Optimization of Dynamic Systems, L. C. W. Dixon and G. P. Szego editors, North Holland, Amsterdam, pp. 3-28.

Eason E. D., and R. G. Fenton, 1974, "A Comparison of Numerical Optimization Methods for Engineering Design," in ASME Journal of Engineering Industry Series B, Vol 96 No 1, pp 196-200, February.

Ecker J. G. and Kupferschmid 1985, "A Computational Comparison of the Ellipsoid Algorithm with Several Nonlinear Programming Algorithms," in SIAM Journal of Control and Optimization, Vol 23, No. 5, September 1985.

Falk, J. E., and Fiacco, A. V., 1982, "The use of Mathematical Programming: Who let the Man Out:", in Computers and Operations Research, Vol. 9, No. 1, pp. 3-5.

Fiacco, A. V., 1976, "Sensitivity Analysis For Nonlinear Programming Using Penalty Methods", in Mathematical Programming, Vol. 10, pp 287-311.

Fiacco, A. V., 1980, "Nonlinear Programming Sensitivity Analysis Results Using Strong Second Order Assumptions", In Numerical Optimization of Dynamic Systems, L. C. W. Dixon and G. P. Szego editors, North Holland, Amsterdam, pp. 349-362.

Fiacco, A. V., 1983, Introduction to Sensitivity and Stability Analysis in Nonlinear Programming, Academic Press, New York.

Fiacco, A. V., and Ghaemi, A., 1982, "Sensitivity Analysis of a Nonlinear Structural Design Problem", in Computers and Operations Research, Vol. 9, No. 1, pp 29-55.

Fiacco A. V. and G. P. McCormick, 1968, Nonlinear Programming: Sequential Unconstrained Programming Techniques, Wiley, New York

Fletcher, R., 1970, "A New Approach to Variable Metric Algorithms," in Computer Journal, Vol 13, pp 317-322.

Fletcher, R., 1985, "Trust Region methods based on the l_1 RQP method," in Nonlinear Optimization 1984, edited by P. T. Boggs, R. H. Byrd, and R. B. Schnable, SIAM Philadelphia, pp 22-40.

Fletcher, R. and Powell, M. J. D., 1974, "On Modification of LDLT Factorizations," in Mathematics of Computation, Vol. 28, No 128, pp 1067-1087.

Fox, R. L., 1971, Optimization Methods for Engineering Design, Addison-Wesley Publishing Company.

Gabriele, G. A., and Beltracchi, T. J., 1986, "A Comparison of Pshenichnyi's Recursive Quadratic Programming Versus Generalized Reduced Gradient Method", in Proceedings of ASME PVP and OR methods in Engineering conference. ASME PVP 109 July 1986.

Gal, T., 1984, "Linear Parametric Programming- A Brief Survey," in Mathematical Programming Study 21, North Holland, pp 43-68.

Ganesh, N. and Biegler, L. T., 1987, "A Reduced Hessian Strategy for Sensitivity Analysis of Optimal Flowsheets," in AIChE Journal February, Vol 33, No. 2, pp 283-296.

Garcia, C. B., and Zangwill, W. I., 1981, Pathways to Solutions, Fixed Points, and Equilibria, Prentice-Hall, New Jersey.

Gauvin, J., and Dubeau, F., 1984, "Some Examples and Counterexamples For The

Stability Analysis of Nonlinear Programming Problems," in Mathematical Programming Study 21, North Holland, pp 69-78.

Gill, P. E. and Murray, W., 1978, "The design and implementation of software for unconstrained optimization," in Design and Implementation of Optimization Software, Ed. by H. Greenberg, Sijthoff and Noordhoff.

Gill, P. E., and Murray, W., 1979, "The Computation of Lagrange Multiplier Estimates for Constrained Minimization," in Mathematical Programming, Vol 17, pp 32-60.

Gill, P. E., Murray, W., and Saunders, M. A., 1975, "Methods For Computing and Modifying the LDV Factors of a Matrix," in Mathematics of Computation, Vol 29, No. 132, pp 1051-1077.

Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., 1983 "Computing Forward-Difference Intervals for Numerical Optimization" in SIAM Journal of Scientific and Statistical Computing, Vol. 4, No. 2, June 1983, pp 310-321.

Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., 1985, "Software and Its Relationship to Methods," in Nonlinear Optimization 1984, edited by P. T. Boggs, R. H. Byrd, and R. B. Schnable, SIAM Philadelphia, pp 139-159.

Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., 1986, "Considerations of Numerical Analysis in a Sequential Quadratic Programming Method", in Numerical Analysis Lecture Notes in Mathematics No. 1230, Springer Verlag, New York 1986.

Gill, P. E., Murray, W., and Wright M. H., 1981, Practical Optimization, Academic Press, London.

Goldfarb, D., 1970, "A Family of Variable Metric Methods Derived by Variational Means," in Mathematics of Computation, Vol 24, pp 23-26.

Guang-Yuan, W., and Wen-Quan, W., 1985, "Fuzzy Optimum Design of Structures", in Engineering Optimizatoin, vol 8 pp. 291-300).

Haftka, R. T., Iott, J., and Adelman, H. M., 1985, "Selecting step sizes in sensitivity analysis by finite differences," Aug. 1985, NASA TM 86382.

Haftka, R. T., and Thareja, R., 1986, "Numerical Difficulties Associated with Using Equality Constraints to Achieve Multi-Level Decomposition in Structural Optimization" AIAA paper 86-0854.

Himmelblau, D. M. 1972, Applied Nonlinear Programming, McGraw Hill.

Haug, E. J., and Arora, J. S., 1979, Applied Optimal Design, Wiley, New York.

Haug, E. J., Komkov, V., and Choi, K. K., 1985, Design Sensitivity Analysis of Structural Systems, Academic Press, Orlando, Fl.

Ip, C. M., 1987, "On Least-Change Secant Updated in Factored Form," In SIAM Journal of Numerical Analysis, Vol 24, No. 5 pp. 1126-1132, October 1987.

Janin, R., 1984, "Directional Derivative of the Marginal Function in Nonlinear Programming," in Mathematical Programming Study 21, North Holland, pp 110-126.

Jittorntrum, K., 1984, "Solution Point Differentiability without Strict

Complementarity in Nonlinear Programming," in Mathematical Programming Study 21, North Holland Amsterdam pp 127-138.

Kojima, M, and Hirabayashi, R., 1984, "Continuous Deformation of Nonlinear Programs," in Mathematical Programming Study 21, North Holland pp 150-198.

Kuhn, H. W., and Tucker, A. W., 1951, "Nonlinear Programming," in Proceedings 2nd Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman (Ed.), University of California Press, Berkeley, California.

Kwak, B. M. and Haug, E. J. Jr., 1976, "Optimum Design in the Presence of Parametric Uncertainty", Journal of Optimization: Theory and Applications, Vol. 19, No. 4, Aug., pp. 527-546.

Lagrange, J. L., 1881, Oeuvres de Lagrange, Paris

McKeown, J. J., 1980 a, "An Approach to Sensitivity Analysis", In Numerical Optimization of Dynamic Systems, L. C. W. Dixon and G. P. Szego editors, North Holland, Amsterdam, pp. 349-362

McKeown, J. J., 1980 b, "Parametric Sensitivity Analysis of Nonlinear Programming Problems", In Nonlinear Optimization Theory and Algorithms, L. C. W. Dixon, E. Spedicato and G. P. Szego editors, North Holland, Amsterdam, pp. 387-406

McKeown, J. J., 1980 c, "Sensitivity Analysis with Respect to Independent Variables", In Nonlinear Optimization Theory and Algorithms, L. C. W. Dixon, E. Spedicato and G. P. Szego editors, North Holland, Amsterdam, pp. 407-427

Murray, W., and Wright, M. H., 1982, "Computation of the Search Direction in Constrained Optimization Algorithms", in Mathematical Programming Study 16, North Holland Publishing Company, pp 62-83.

Oren, 1974, "Self Scaling Variable Metric (SSVM) Algorithms, Part 2: Implementation and Experiments," in Management Science Vol 20, pp 863-874.

Padula, S. L., and Young, K. C., 1986, "Simulator for Multilevel Optimization Research," NASA TM-87751, June 1986.

Padula, S. L., and Sobieszczanski-Sobieski, J., 1987, "A Computer Simulator for Development of Engineering System Design Methodologies," NASA TM-89109, February 1987.

Powell, M. J. D., 1977, "A fast algorithm for nonlinearly constrained optimization calculation," in ed G. A. Watson, Numerical Analysis Dundee, Lecture Notes in Mathematics 630, Springer Verlag, Berlin 1978, pp 144-157.

Powell, M. J. D., 1978, "The Convergence of Variable Metric Methods for nonlinearly constrained optimization calculations," in Nonlinear Programming 3, ed. by O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, Academic Press, London and New York, pp 27-63.

Powell, M. J. D., 1981, "Nonlinear Optimization 1981", Chapters 5 comments 5.22, 5.23, 5.24, and Chapter 6 comment 6.16, pp 324-326, 400.

Powell, M. J. D., 1983, "Variable Metric Methods for Constrained Optimization," in Mathematical Programming, The State of The Art Bonn 1982, Ed. by A. Bachem, M. Grotschel, and B. Korte, Springer Verlag.

Powell, M. J. D., 1985, "A comparison of two Subroutines on some Difficult test Problems," in Nonlinear Optimization 1984, ed P. T. Boggs, R. H. Byrd, and R. B. Schnable, SIAM Philadelphia pp 160-178.

Powell, M. J. D., 1986, "How Bad are the BFGS and DFP Methods when the Objective Function is Quadratic?", in Mathematical Programming vol 34. pp 34-47.

Rao, S. S., 1987 a, "Description and Optimum Design of Fuzzy Mechanical Systems", to appear in ASME Journal of Mechanisms, Transmissions, and Automation in Design.

Rao, S. S., 1987 b, "Multi-Objective Optimization of Fuzzy Structural Systems," in International Journal for Numerical Methods in Engineering, June 1987, pp 1157-1171.

Reklaitis, G. V., Ravindran, A., Ragsdell, K. M., 1983, Engineering Optimization Methods and Applications, Wiley, New York.

Robertson, W. D., and Gabriele, G. A., 1987, "The Optimal Design of a Rotary Type Actuator for Magnetic Disk Drives," in Advances in Design Automation - 1987, Volume One: Design Methods, Computer Graphics, and Expert Systems, ASME DE-Vol. 10-1, pp 107-114, to Appear in ASME Journal of Mechanisms, Transmissions, and Automation in Design.

Rockafellar, R. T., 1984, "Directional Differentiability of the Optimal Value Function in a Nonlinear Programming Problem," in Mathematical Programming Study 21, North Holland, pp 213-226.

Rosen, J. B., and Suzuki S., 1965, "Construction of Nonlinear Programming Test Problems," in Communications of the ACM, vol 8 no 2, p 113.

Samuelson, P. A., 1947, Foundations of Economic Analysis, Harvard University Press, Cambridge Mass.

Sandgren E., 1977, "The Utility of Nonlinear Programming Algorithms", Ph.D. dissertation Purdue University Dec. 1977, West Lafayette In.

Sandgren, E., Gim, G., and Ragsdell, K. M., 1985, 'Optimal Design of a Class of Welded Beam Structures Based on Design for Latatude," in ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol 107 Dec. pp 482-487.

Schittkowski, K., 1980, Nonlinear Programming Codes: Information, Tests, Performance, Lecture Notes in Economic and Mathematical Systems, Vol 183, Springer Verlag, New York.

Schittkowski, K., 1983, "On the Convergence of a Sequential Quadratic Programming method with an Augmented Lagrangian Line Search Function," in Mathematisch Operationsforsch. Statist. Ser. Optim. Vol 14 no. 2, pp 197-216.

Schmit, L. A., and Chang, K. J., 1984, "Optimum Design Sensitivity Based on Approximation Concepts and Dual Methods", in International Journal for Numerical Methods in Engineering, vol. 20, pp 39-75.

Shanno, D. F., 1970, "Conditioning of Quasi Newton Methods for Function Minimization," in Mathematics of Computation, Vol 24, pp 647-657.

Shanno, D. F., and Phua, K. H., 1978 a, "Numerical Comparison of Several Variable Metric Algorithms," in Journal of Optimization Theory and Applications, Vol 25, No 4, Aug., pp 507-518.

Shanno, D. F., and Phua, K. H., 1978 b, "Matrix Conditioning and Nonlinear Optimization," in Mathematical Programming vol 14, pp 149-160.

Sobieszczanski-Sobieski, J., 1982, A Linear Decomposition Method for Large Optimization Problems --Blueprint for Development, NASA TM 83248 Feb. 1982.

Sobieski, J., 1984, "Multidisciplinary Systems Optimization By Linear Decomposition" in Recent Experiences in Multidisciplinary Analysis and Optimization, Part 1, Compiled by J. Sobieski, NASA CP 2327, pp 343-366.

Sobieszczanski-Sobieski, J., 1986, "The Case for Aerodynamic Sensitivity Analysis," in Sensitivity Analysis in Engineering, Edited by H. M. Alderman and R. T. Haftka, NASA CP-2457, pp 77-93.

Sobieszczanski-Sobieski, J., and Barthelemy, J. F., 1985, "Improving Engineering System Design by Formal Decomposition, Sensitivity Analysis, and Optimization", NASA TM 86377 Feb 1985.

Sobieszczanski-Sobieski, J., Barthelemy, J. F., and Riley K. M., 1981, "Sensitivity of Optimum Solutions to Problem Parameters". Proceedings of the AIAA/ASME/ASCE/AHS 22nd Structures, Structural Dynamics and Materials Conference, Atlanta, Ga., April 1981, pp 184-205, also NASA TM 83134, May 1981, AIAA Journal, Vol 20., Sept. 1982, pp. 1291-1299.

Sobieszczanski-Sobieski, J., James, B. B., and Dowi, A. R., 1985, "Structural Optimization by Multilevel Decomposition", in AIAA Journal Vol 23, no. 11.

Sobieszczanski-Sobieski, J., James, B. B., and Riley, M. F., 1987, "Structural Sizing be Generalized, Multilevel Optimization," in AIAA Journal Vol. 25, No. 1, January 1987, pp 139-145

Templeman, A. B., 1987, "Computer Aided Optimum Structural Design Under Uncertainty," in Engineering Optimization, Vol 11, pp 281-288

Van der Hoek, G., 1980 "Reduction methods in Nonlinear Programming" Mathematical Centre Tracts 126, Mathematisch Centrum, Amsterdam 1980.

Vanderplaats, G. N., 1984 a, "An Efficient Feasible Directions Algorithm for Design Synthesis", in AIAA Journal, Vol. 22, No. 11, Nov. , pp. 1633-1639

Vanderplaats, G. N., 1984 b, Numerical Optimization Techniques for Engineering Design: with Applications, McGraw-Hill Book Co., New York.

Vanderplaats, G. N., and Cai, H. D., 1987, "Alternative Methods for Calculating Sensitivity of Optimized Designs to Problem Parameters," in Sensitivity Analysis in Engineering, Edited by H. M. Alderman and R. T. Haftka, NASA CP-2457, pp 19-32.

Vanderplaats, G. N., and Yoshida, N., 1985, "Efficient Calculation of Optimum Design Sensitivity" in AIAA Journal, Nov. , pp 1798-1803

Zedah, L., 1965, "Fuzzy Sets," Information and Control, Vol 9, pp. 338-353.

Zhou J., and Mayne, R. W. 1985,, "Monotonicity Analysis and Recursive Quadratic Programming in Constrained Optimization," in ASME Journal of Mechanisms, Transmissions, and Automation in Design, December Vol 107, pp 459-462.

Zolezzi, T., 1985, "Continuity of Generalized Gradients and Multipliers Under

Perturbations," in Mathematics of Operations Research, Vol. 10, No. 4, November, pp 664-673